



TUGAS AKHIR – SS 141501

**ANALISIS SENTIMEN PENGGUNA *TWITTER*
TERHADAP PEMILIHAN GUBENUR DKI
JAKARTA DENGAN METODE *NAÏVE BAYESIAN*
CLASSIFICATION DAN *SUPPORT VECTOR*
*MACHINE***

**EZA PUTRA NUANSA
NRP 1313 100 114**

**Dosen Pembimbing
Dr. Kartika Fithriasari, M.Si**

**PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**



TUGAS AKHIR – SS 141501

**ANALISIS SENTIMEN PENGGUNA *TWITTER*
TERHADAP PEMILIHAN GUBERNUR DKI
JAKARTA DENGAN METODE *NAÏVE BAYESIAN*
CLASSIFICATION DAN *SUPPORT VECTOR*
*MACHINE***

**EZA PUTRA NUANSA
NRP 1313 100 114**

**Dosen Pembimbing
Dr. Kartika Fithriasari, M.Si**

**PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**



FINAL PROJECT – SS 141501

**SENTIMENT ANALYSIS FOR TWITTER USER TO
ELECTION DKI JAKARTA GOVERNOR USING
NAÏVE BAYESSIAN CLASSIFICATION AND
SUPPORT VECTOR MACHINE METHOD**

**EZA PUTRA NUANSA
NRP 1313 100 114**

**Supervisor
Dr. Kartika Fithriasari, M.Si**

**UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**

LEMBAR PENGESAHAN

ANALISIS SENTIMEN PENGGUNA *TWITTER* TERHADAP PEMILIHAN GUBERNUR DKI JAKARTA DENGAN METODE *NAÏVE BAYESIAN* *CLASSIFICATION* DAN *SUPPORT VECTOR MACHINE*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
pada

Program Studi Sarjana Departemen Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember

Oleh :

Eza Putra Nuansa

NRP. 1313 100 114

Disetujui oleh Pembimbing:

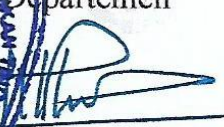
Dr. Kartika Fithriasari, M.Si

NIP. 19691212 199503 2 002

()



Mengetahui,
Kepada Departemen



Dr. Buhartono

NIP. 19710929 199512 1 001

SURABAYA, JULI 2017

ANALISIS SENTIMEN PENGGUNA *TWITTER* TERHADAP PEMILIHAN GUBERNUR DKI JAKARTA DENGAN METODE *NAÏVE BAYESIAN* *CLASSIFICATION* DAN *SUPPORT VECTOR MACHINE*

Nama Mahasiswa : Eza Putra Nuansa
NRP : 1313 100 114
Departemen : Statistika
Dosen Pembimbing : Dr. Kartika Fithriasari, M.Si

Abstrak

Pada tahun 2017, DKI Jakarta akan melakukan pesta demokrasi yang tentunya berpengaruh terhadap dunia social media. Sayangnya, sebagian besar perbincangan di Twitter itu adalah bentuk serangan verbal yang sering menggunakan kata-kata kasar dan menghembuskan isu sensitif seperti agama serta etnis untuk menyerang kandidat lain. Sehingga, twitter sangat cocok untuk dijadikan sumber analisa sentimen dan opinion mining karena penggunaan Twitter untuk mengekspresikan opini mereka terhadap berbagai topik.

Penelitian ini bertujuan untuk mencari kata kunci dan hubungan pola antar tiap calon Gubernur DKI Jakarta. Metode Naive Bayes Classifier pada masing-masing pengukuran performa akurasi, precision, recall, dan F-Measure sebesar 85.77%; 85.90%; 85.77%; 85.67%. Metode Support Vector Machine kernel RBF tiap pengukuran performa akurasi, precision, recall, dan F-Measure adalah 87.80%; 98.48%; 87.80%; 92.64%. Untuk hasil SNA didapatkan hasil yang tinggi untuk degree centrality sebesar 0.865 yang menunjukkan pengaruh antar node kata kunci dengan kata kunci yang lain.

Kata Kunci : Analisis Sentimen, Naïve Bayes Classification, Pemilihan Gubernur DKI Jakarta, Social Network Analysis, Support Vector Machine.

(halaman ini sengaja dikosongkan)

SENTIMENT ANALYSIS FOR TWITTER USER TO ELECTION DKI JAKARTA GOVERNOR USING NAÏVE BAYESSIAN CLASSIFICATION AND SUPPORT VECTOR MACHINE METHOD

Student Name : Eza Putra Nuansa
Student Number : 1313 100 114
Department : Statistics
Supervisor : Dr. Kartika Fithriasari, M.Si

Abstract

In 2017, DKI Jakarta will celebrate a democracy party which certainly affects the netizen in social media. Unfortunately, most of the conversation on Twitter is a form of verbal attack that often uses harsh words and exhaling sensitive issues such as religion and ethnicity to attack other candidates. Thus, twitter is perfect for being a source of sentimental and opinion mining analysis because of the use of Twitter to express their opinions on various topics.

This study to find the keyword and pattern relationship between each candidate of Governor of DKI Jakarta. Method of Naive Bayes Classifier in each measurement of accuracy, precision, recall, and F-Measure of 85.77%; 85.90%; 85.77%; 85.67%. Support Method Vector Machine RBF kernel for each measurement of accuracy, precision, recall, and F-Measure performance is 87.80%; 98.48%; 87.80%; 92.64%. For Social Network Analysis results obtained high results for degree centrality 0.865 which shows the influence of keyword nodes with other keywords.

Key words : Election Govenor DKI Jakarta, Naïve Bayes Classification, Sentiment Analysis, Social Network Analysis, Support Vector Machine

(halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur yang kehadirat Allah SWT, Tuhan Yang Maha Esa. Berkat rahmat Nya penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“ANALISIS SENTIMEN PENGGUNA TWITTER TERHADAP PEMILIHAN GUBERNUR DKI JAKARTA DENGAN METODE NAIVE BAYESIAN CLASSIFICATION DAN SUPPORT VECTOR MACHINE”** dengan lancar.

Keberhasilan penyusunan Tugas Akhir ini tidak lepas dari bantuan dan dukungan yang diberikan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Suhartono selaku Ketua Departemen Statistika dan Bapak Dr. Sutikno selaku Ketua Program Studi S1 yang telah memberikan fasilitas untuk kelancaran penyelesaian Tugas Akhir.
2. Ibu Dr. Kartika Fithriasari, M.Si selaku dosen pembimbing dalam memberikan bimbingan, selama penyusunan Tugas Akhir.
3. Bapak Dr. Bambang Widjarnako Otok, S.Si, M.Si dan Ibu Dra. Wiwiek Setya Winahju, M.S selaku dosen penguji yang telah memberikan bimbingan untuk Tugas Akhir ini.
4. Kedua orang tua yang selalu memberikan dukungan, serta pemicu bagi penulis dalam menyelesaikan Tugas Akhir.
5. Semua pihak yang telah memberikan bantuan hingga penyusunan laporan Tugas Akhir ini dapat terselesaikan.

Dengan selesainya laporan ini, Penulis berharap hasil Tugas Akhir ini dapat bermanfaat bagi kita semua.

Surabaya, Mei 2017

Penulis

(halaman ini sengaja dikosongkan)

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xiv
 BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	6
1.5 Batasan Penelitian	6
 BAB II TINJAUAN PUSTAKA	
2.1 <i>Teks Mining</i>	7
2.2 Analisis Sentimen	8
2.3 Klasifikasi Teks	9
2.4 Ketepatan Klasifikasi Model	9
2.5 <i>Confix-Stripping Stemmer</i>	10
2.6 <i>Naïve Bayes Classifier</i>	12
2.7 <i>Term Frequency Inverse Document Frequency</i>	16
2.8 <i>Support Vector Machine</i>	17
2.8.1 <i>Support Vector Machine Linier</i>	17
2.8.2 <i>Support Vector Classification</i>	18
2.8.2 <i>Fungsi Kernel pada SVM</i>	19
2.9 Pengukuran Performa Klasifikasi	23
2.10 <i>Twitter</i>	24
2.11 Pemilihan Kepala Daerah DKI Jakarta 2017	24

BAB III METODOLOGI PENELITIAN

3.1 Sumber Data	27
3.2 Struktur Data.....	27
3.3 Langkah Analisis	30
3.4 Diagram Alir.....	33

BAB IV ANALISIS DAN PEMBAHASAN

4.1 Karakteristik Data <i>Tweet</i> Berdasarkan calon Gubernur DKI Jakarta.....	35
4.2 Praproses Teks.....	37
4.3 <i>Naïve Bayes Classification</i>	42
4.3.1 Pengukuran Performa NBC dengan Data <i>Tweet</i> 10-cross validation	43
4.4 <i>Support Vector Machine</i>	46
4.4.1 SVM menggunakan Kernel RBF pada Data <i>Tweet</i>	47
4.4.2 SVM menggunakan Kernel Linier pada Data <i>Tweet</i>	49
4.4.3 Penentuan Ketepatan Klasifikasi Sentimen dengan Kernel RBF dan Linier	50
4.4.4 Pengukuran Performa SVM	51
4.5 Perbandingan Hasil Klasifikasi antara NBC dan SVM.....	55
4.6 Hubungan antar Kata Kunci Ketiga Calon Gubernur DKI Jakarta.....	55
4.3.5 <i>Wordcloud</i> Interaksi antar Ketiga Calon Gubernur DKI Jakarta	55
4.4 <i>Social Network Analysis</i>	58

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan.....	61
5.2 Saran	62

DAFTAR PUSTAKA	63
-----------------------------	----

LAMPIRAN	67
-----------------------	----

BIODATA PENULIS

DAFTAR TABEL

	Halaman
Tabel 2.1 Ilustrasi hasil pembobotan kata kunci	12
Tabel 2.2 Ilustrasi perhitungan NBC pada sentimen positif.....	15
Tabel 2.3 Ilustrasi probabilitas NBC pada sentimen positif.....	15
Tabel 2.4 Ilustrasi perhitungan NBC pada sentimen negatif.....	15
Tabel 2.5 Ilustrasi probabilitas NBC pada sentimen negatif.....	16
Tabel 2.6 Fungsi Kernel Umum SVM	21
Tabel 2.7 Ilustrasi <i>tweet</i> yang umum pada SVM.....	22
Tabel 2.8 Ilustrasi <i>tdf-idf</i> (jumlah bobot keseluruhan per-kata).....	24
Tabel 2.9 Ilustrasi <i>tweet</i> dalam menentukan <i>support vector</i>	23
Tabel 3.1 Struktur Data sebelum <i>preprocessing</i>	27
Tabel 3.2 Struktur Data setelah <i>preprocessing</i>	29
Tabel 4.1 Contoh praproses teks	38
Tabel 4.2 Beberapa Contoh <i>Stopword</i> yang di gunakan ...	38
Tabel 4.3 Frekuensi kata kunci calon Gubernur DKI Jakarta Agus	39
Tabel 4.4 Frekuensi kata kunci calon Gubernur DKI Jakarta Ahok	39
Tabel 4.5 Frekuensi kata kunci calon Gubernur DKI Jakarta Anies.....	40

Tabel 4.6	Frekuensi kata kunci tiap calon Gubernur DKI Jakarta	41
Tabel 4.7	Frekuensi kata kunci ketiga calon Gubernur DKI Jakarta	41
Tabel 4.8	Ketepatan Klasifikasi Pembentukan Model NBC Menggunakan Data <i>Tweet</i>	42
Tabel 4.9	Hasil Akurasi, <i>Precision</i> , <i>Recall</i> , dan <i>F-Measure</i> NBC pada Data <i>Tweet</i>	44
Tabel 4.10	Hasil <i>confusion matrix</i> NBC pada Data <i>Tweet</i> .	44
Tabel 4.11	Hasil Akurasi, <i>Precision</i> , <i>Recall</i> , dan <i>F-Measure</i> NBC pada Data <i>Tweet</i>	45
Tabel 4.12	Hasil <i>confusion matrix</i> NBC pada Data <i>Tweet</i> .	45
Tabel 4.13	Prediksi <i>tweet</i> antar Calon Gubernur DKI Jakarta	46
Tabel 4.14	Ketepatan Klasifikasi SVM Kernel RBF Menggunakan Data <i>Tweet</i>	47
Tabel 4.15	Ketepatan Klasifikasi Pembentukan Model NBC Menggunakan Data <i>Tweet</i>	49
Tabel 4.16	Ketepatan Klasifikasi SVM Kernel RBF Menggunakan Data <i>Tweet</i>	50
Tabel 4.17	Ketepatan Klasifikasi Sentimen SVM Kernel RBF dan Linier	50
Tabel 4.18	Hasil Akurasi, <i>Precision</i> , <i>Recall</i> , dan <i>F-Measure</i> SVM dengan Kernel RBF pada Data <i>Tweet</i>	52
Tabel 4.19	Hasil <i>confusion matrix</i> NBC pada Data <i>Tweet</i> .	52
Tabel 4.20	Hasil Akurasi, <i>Precision</i> , <i>Recall</i> , dan <i>F-Measure</i> SVM pada Data <i>Tweet</i>	53
Tabel 4.20	Persamaan Model SVM Kernel RBF	54
Tabel 4.22	Perbandingan Metode NBC dan SVM.....	55

Tabel 4.23	Deskripsi <i>SNA</i> tipe <i>Fruchterman-Reingold</i>	60
-------------------	---	----

DAFTAR GAMBAR

	Halaman
Gambar 2.1	Contoh hasil <i>crawling</i> data dari <i>Twitter API</i> ... 11
Gambar 2.2	Contoh hasil <i>case folding</i> 11
Gambar 2.3	Contoh hasil <i>cleansing data</i> 11
Gambar 2.4	Contoh hasil <i>steeming</i> 12
Gambar 2.5	Ilustrasi <i>Hyperplane</i> pada Metode SVM..... 17
Gambar 2.6	Fungsi memetakan data 20
Gambar 3.1	Diagram Alir Praproses Teks 33
Gambar 3.2	Diagram Alir Membangun model NBC dan SVM 34
Gambar 4.1	Karakteristik data <i>tweet</i> calon Gubernur DKI Jakarta Agus Yudhoyono 35
Gambar 4.2	Karakteristik data <i>tweet</i> calon Gubernur DKI Jakarta Ahok..... 36
Gambar 4.3	Karakteristik data <i>tweet</i> calon Gubernur DKI Jakarta Anies Baswedan..... 37
Gambar 4.4	Persentase Akurasi Data <i>Tweet</i> NBC 43
Gambar 4.5	Ketepatan Klasifikasi SVM Kernel RBF menggunakan data <i>tweet</i> 48
Gambar 4.6	Penentuan Ketepatan Klasifikasi SVM Kernel RBF dan Linier..... 51

Gambar 4.7 Ilustrasi SVM Kernel RBF ketiga calon Gubernur DKI Jakarta	54
Gambar 4.8 <i>Wordcloud</i> Ahok – Anies	56
Gambar 4.9 <i>Wordcloud</i> Anies - Agus	56
Gambar 4.10 <i>Wordcloud</i> Ahok - Agus	57
Gambar 4.11 <i>Wordcloud</i> Ahok – Agus – Anies	58
Gambar 4.12 <i>SNA</i> Ahok – Agus - Anies.....	59

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Data <i>Tweet</i> Asli dari <i>Twitter API</i> (Contoh Data <i>Tweet</i> Ahok).....	67
Lampiran 2 Data <i>Tweet</i> Setelah Praproses Teks (Contoh Data <i>Tweet</i> Ahok)	68
Lampiran 3 <i>Syntax Pre-processing Text with Python</i>	69
Lampiran 4 Frekuensi Kata Kunci yang sering muncul	70
Lampiran 5 <i>Syntax Python</i> mencari frekuensi dokumen	71
Lampiran 6 <i>Syntax Python</i> SVM Kernel RBF	71
Lampiran 7 <i>Syntax Wordcloud</i> di <i>Python</i>	74
Lampiran 8 Hasil <i>Social Network Analysis</i> menggunakan <i>Gephi 0.9.1</i>	75
Lampiran 9 <i>Output Prediksi Naïve Bayes Classification</i> ketiga calon Gubernur DKI Jakarta	75
Lampiran 10 <i>Output Prediksi Support Vector Machine</i> ketiga calon Gubernur DKI Jakarta	77
Lampiran 11 Surat Pernyataan Penggunaan Data	79

(halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Media jejaring sosial memberikan peran yang sangat besar khususnya perkembangan teknologi dalam bidang komunikasi yang menjadi tak terbatas. *Twitter* merupakan sebuah situs *micro-blogging* yang sangat populer di Indonesia. Hal ini terlihat dari jumlah pengguna *Twitter* yang mencapai 19,5 juta pengguna dari total 330 juta pengguna di dunia (Wicaksono, 2014). Dengan tampilan yang mudah penggunaannya, berisi maksimal 140 karakter masyarakat umum bisa menyebutkan dengan kata “*tweet*” atau kicauan. Kata yang terkandung dalam *Twitter* adalah bahasa alami manusia yang merupakan bahasa dengan struktur kompleks. Kebiasaan masyarakat mengutarakan pendapatnya melalui media sosial terutama *Twitter* dalam menanggapi kejadian atau hal-hal yang terjadi di lingkungannya dapat menjadi salah satu acuan untuk mengetahui sentimen masyarakat terhadap lingkungan atau kota tempat tinggal masyarakat tersebut berupa kritik atau saran (Arifiyanti, 2014). Salah satunya menanggapi berbagai macam sentimen masyarakat terhadap Pilkada serentak gelombang kedua yang diselenggarakan bulan Februari 2017.

Komisi Pemilihan Umum (KPU) telah menetapkan 101 daerah menyelenggarakan pemilihan kepala daerah (Pilkada) serentak 2017. Jumlah itu terdiri dari 7 provinsi, 18 kota serta 76 kabupaten. Ibukota Jakarta pastinya akan menjadi sorotan utama dalam Pilkada serentak tahun ini di berbagai media massa dan menjadi topik yang hangat diperbincangkan juga di media sosial. Percakapan-percakapan di media sosial ini mengandung sentimen berupa persepsi para pengguna internet atau *netizen* terhadap pasangan calon, baik itu sentimen positif maupun negatif (Hidayat, 2014). Menurut hitungan lembaga pemantau dan riset media sosial, *Politicawave* sejak bulan Agustus silam, perbincangan tentang calon kandidat di media sosial, khususnya *Twitter*, bisa mencapai rata-rata 20.000 hingga 40.000 perhari. Kebisingan di dunia maya

ini, menurut Saraswati (2011), yang membuat pemilihan Gubernur DKI Jakarta menjadi menarik untuk diamati. Sayangnya, sebagian besar perbincangan di *Twitter* itu adalah bentuk serangan verbal yang sering meng-unakan kata-kata kasar dan menghembuskan isu sensitif seperti agama serta etnis untuk menyerang kandidat lain. Sehingga, *twitter* sangat cocok untuk dijadikan sumber analisa sentimen dan *opinion mining* karena penggunaan *Twitter* untuk mengeks-presikan opini mereka terhadap berbagai topik. Kemudian pengguna *Twitter* juga bervariasi dari pengguna biasa hingga selebritis, perwakilan perusahaan, politikus bahkan presiden.

Keberadaan *twitter* telah digunakan secara luas dari lapisan masyarakat dapat dilihat sebagai sebuah refleksi yang baik di mana keberadaan *Twitter* dapat merepresentasikan apa yang sedang menjadi *trend* pembicaraan dan hal apa yang sedang menarik untuk dibahas. Kebiasaan masyarakat mem-*posting tweet* dalam menilai tokoh publik dapat menjadi acuan untuk menge-tahui sentimen masyarakat terhadap tokoh publik. Kebutuhan dari analisis sentimen terhadap tokoh publik biasanya datang apabila ada pihak yang ingin mengetahui sentimen dan tanggapan publik menjelang pemilihan umum. Kebutuhan tersebut biasanya dimiliki oleh tokoh publik, atau khusus pada tokoh politik seperti calon gubernur, presiden, menteri, atau ketua partai. Oleh karena itu, analisis sentimen terhadap tokoh publik dari *Twitter* menjelang adanya pemilihan umum sangat bermanfaat dalam mem-berikan tambahan wawasan serta gambaran bagi masyarakat tentang tokoh publik yang menjadi kandidat dalam pemilihan umum (Kurniawan, 2012).

Pada dasarnya, setiap proses utama akan melakukan tahapan secara umum yaitu ketika peneliti ingin menentukan orientasi sentimen masyarakat dari *twitter*. Peneliti akan melakukan *input* berupa teks *twitter* atau yang biasa disebut dengan *tweet*. Selanjutnya akan dilakukan seleksi *tweet* yang telah di *input* peneliti sebagai bagian dari proses penentuan sentimen yang mengandung *tweet* positif atau negatif secara manual. Proses ini disebut dengan *Golden Training*, yaitu penentuan positif atau

negatif berdasarkan dari persepsi dari peneliti setelah dilakukan survey kepada beberapa orang untuk menyamakan persepsi peneliti dalam melakukan penilaian tweet yang mengandung positif atau negatif secara manual.

Sebelum *tweet* dapat dikategorikan, maka data *tweet* tersebut sebaiknya diproses terlebih dahulu. Bentuk data yang tersedia pada *tweet* adalah berupa teks. Apabila dibandingkan dengan jenis data yang lain, sifat data berbentuk teks tidak terstruktur dan sulit untuk ditangani. *Text mining* adalah cara agar teks diolah dengan menggunakan komputer untuk menghasilkan analisis yang bermanfaat (Witten, Frank, & Hall, 2011). Praproses dalam *text mining* diantaranya adalah *tokenizing*, *case folding*, *stopwords*, dan *stemming*. Diantara keempat langkah tersebut yang penting adalah proses *stemming*. *Stemming* merupakan proses menghilangkan imbuhan pada suatu kata untuk mendapatkan kata dasar dari kata tersebut. Tidak seperti bahasa Inggris yang imbuhan ada pada akhiran, imbuhan pada bahasa Indonesia lebih rumit, yakni awalan, akhiran, sisipan, dan *confixes* kom-binasi awalan dan akhiran. Hal ini dapat ditangani oleh *confix-stripping stemmer* dengan mengubah urutan pemecahan imbuhan pada beberapa jenis imbuhan tertentu atau *rule precedence*. Setelah itu penggunaan *stopwords* akan menjadi salah satu perhatian. *Stopwords* merupakan kosakata yang bukan kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada teks atau kalimat. Diharapkan melalui *text processing* data telah siap untuk diolah lebih lanjut.

Salah satu metode statistika yang dapat melakukan pengkategorian adalah klasifikasi. Klasifikasi merupakan suatu metode untuk memprediksi kategori kelas dari suatu data. Beberapa metode klasifikasi cukup banyak digunakan untuk melakukan klasifikasi berupa teks. Metode klasifikasi tersebut diantaranya adalah *Naïve Bayes Classifier* (NBC), *K-Nearest Neighbour*, dan *Support Vector Machines* (SVM). Penelitian ini akan menggunakan metode NBC dan SVM. Metode NBC telah banyak digunakan dalam penelitian mengenai *text mining*, beberapa

kelebihan NBC diantaranya adalah salah satu algoritma klasifikasi yang sederhana namun memiliki akurasi yang tinggi (Rish, 2006). Sedangkan pemilihan metode SVM karena kemampuannya generalisasi dalam mengklasifikasikan suatu *pattern* / pola. Teknik ini berakar pada teori pembelajaran statistik dan telah menunjukkan hasil empiris yang menjanjikan dalam berbagai aplikasi praktis dari pengenalan digit tulisan tangan sampai kategorisasi teks. SVM juga bekerja sangat baik pada data dengan berbagai banyak dimensi dan menghindari kesulitan dari permasalahan dimensionalitas (Tan, Steinbach, & Kumar, 2006).

Penelitian berkaitan dengan metode NBC dan SVM untuk kasus data *tweet* telah dilakukan diantaranya oleh Sunni dan Hanifah (2012) mengulas analisis sentimen menggunakan algoritma Naïve Bayes dengan penggabungan 10 metode praproses dan melihat karakter topik yang muncul menggunakan metode Tf-Idf. Data yang digunakan yaitu sebanyak 2000 tweet dan didapatkan hasil yaitu tingkat akurasi antara 69,4-72,8%. Selanjutnya dari Pak dan Paroubek (2010) tentang penggunaan media twitter sebagai bahan untuk analisis sentimen dan pertimbangan opini dengan cara klasifikasi menggunakan algoritma Naïve Bayes dan mencoba melakukannya dengan metode lain yaitu SVM dan CRF. Data yang digunakan yaitu sebanyak 216 tweet dari akun twitter agensi koran dan majalah di US yang dibagi ke dalam 3 kelas yaitu mengandung emosi positif, negatif, dan tidak menunjukkan emosi apapun dan didapatkan hasil yaitu tingkat akurasi berada pada nilai $\pm 80\%$ dengan fitur unigram, bigram dan trigram. Dalam mengulas tentang penerapan analisis sentimen pada twitter berbahasa Indonesia sebagai pemberi rating dengan menggunakan algoritma SVM dan proses stemming oleh Monarizqa, dkk (2014). Data yang digunakan yaitu sebanyak 175.000 tweet yang dibagi ke dalam 2 kelas yaitu tweet positif dan negatif. Hasil terbaik yang didapatkan yaitu tingkat akurasi sebesar 73,43%. Terdapat penelitian sebelumnya yang juga menggunakan 2 algoritma yaitu NBC dan SVM. Pada penelitian ini dilakukan pada pengayaan teks melalui Wikitology oleh Hassan, dkk (2012) dengan menggunakan data

dari 20 kelompok berita dan seimbang di setiap kategorinya yaitu 1000. Hasil yang didapat menunjukkan bahwa penggunaan algoritma NBC dengan validasi 10 fold cross validation adalah adanya penambahan sebesar 28,78% dan SVM 6,36%.

1.2 Perumusan Masalah

Dengan latar belakang yang sudah dipaparkan, pada bagian ini akan dituliskan rumusan untuk klasifikasi sentimen pada *tweet* terhadap pemilihan Gubernur DKI Jakarta agar diketahui sebuah *tweet* mempunyai tipe kalimat opini yang positif dan negatif. Untuk itu dalam penelitian ini akan dilakukan analisis sentimen pengguna *twitter* terhadap pemilihan Gubernur DKI Jakarta menggunakan klasifikasi dengan Metode *Naive Bayes Classifier* dengan *Support Vector Machine* dalam mencari kata kunci dari ke tiga calon Gubernur dan hubungan antar setiap calon Gubernur. Pemilihan metode NBC dan SVM digunakan karena dari akurasi keduanya mempunyai hasil yang tinggi.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan yang ingin dicapai dalam penelitian ini adalah

1. Mengetahui karakteristik data yang digunakan berdasarkan data *tweet* calon Gubernur DKI Jakarta.
2. Mengetahui *pra proses* teks pada data *tweet* calon Gubernur DKI Jakarta.
3. Menguji ketepatan data *tweet* prediksi klasifikasi dengan Metode *Naive Bayes Classifier*
4. Mendapatkan model persamaan data *tweet Support Vector Machine*.
5. Mengetahui perbandingan akurasi antara NBC dan SVM
6. Mengetahui hubungan kata kunci cagub DKI Jakarta

1.4 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat bermanfaat dalam bidang klasifikasi *tweet* secara umum dengan menggunakan metode NBC dan SVM. Penelitian ini diharapkan dapat membantu membantu pihak-pihak yang ingin mengetahui kata kunci yang identik terhadap tokoh publik calon Gubernur DKI Jakarta melalui analisis sentimen melalui *posting tweet* masyarakat pengguna *Twitter* serta hubungan antar setiap calon Gubernur.

1.5 Batasan Penelitian

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya melakukan analisis sentimen terhadap *tweet* berbahasa Indonesia.
2. Tokoh yang dinilai dalam penelitian ini hanya dibatasi yaitu tokoh yang maju sebagai calon Gubernur DKI Jakarta yaitu sebanyak 3 calon.
3. Penelitian ini tidak mengatasi kata dan kalimat yang cara penulisannya tidak umum (disingkat).
4. Penelitian ini tidak mengatasi *tweet* yang mempunyai kata atau frasa dengan arti ganda dan berbeda pada sebuah kalimat.
5. Data *tweet* menggunakan periode masa tenang putaran 1 Pilkada Serentak dari 11 Februari 2017 – 14 Februari 2017.

BAB II TINJAUAN PUSTAKA

2.1 *Text Mining*

Text mining merupakan salah satu cabang ilmu *data mining* yang menganalisis suatu data berupa teks. Menurut Han, Kamber, dan Pei (dalam Prilian, 2014), *text mining* adalah suatu langkah analisis teks yang dilakukan otomatis oleh komputer untuk menggali informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen. Ide awal pembuatan *text mining* adalah untuk menemukan pola-pola informasi yang dapat digali dari teks yang tidak terstruktur (Hamzah, 2012). Dengan demikian, *text mining* mengacu juga kepada istilah *text data mining* (Hearst, 1997) atau penemuan pengetahuan dari basis data teks (Feldman dan Dagan dalam Hamzah, 2012). Saat ini, *text mining* telah mendapat perhatian dalam berbagai bidang, antara lain dibidang keamanan, biomedis, pengembangan perangkat lunak dan aplikasi, media *online*, pemasaran, dan akademik. Seperti halnya dalam *data mining*, aplikasi *text mining* pada suatu studi kasus, harus dilakukan sesuai prosedur analisis. Langkah awal sebelum suatu data teks dianalisis menggunakan metode-metode dalam *text mining* adalah melakukan *pre-processing* teks. Sehingga, setelah didapatkan data yang siap diolah, analisis *text mining* dapat dilakukan.

Text mining dapat digunakan untuk proses penemuan *rule* baru dengan algoritma pengelompokan, asosiasi, dan *ranking*. Beberapa fungsi tersebut, yang paling banyak dilakukan adalah proses pengelompokan. Terdapat dua jenis metode pengelompokan teks, yaitu *text clustering* dan *text classification*. Menurut Darujati dan Gumelar di tahun 2012, *text clustering* berhubungan dengan proses menemukan sebuah struktur kelompok yang belum terlihat (tak terpandu atau *unsupervised*) dari sekumpulan dokumen. Sedangkan *text classification* dapat dianggap proses untuk membentuk golongan (kelas-kelas) dari dokumen berdasarkan pada kelas kelompok yang sudah diketahui

sebelumnya (terpandu atau *supervised*). Berdasarkan pengertian ini, dapat dinyatakan bahwa proses klasifikasi (*supervised*) merupakan proses yang lebih mudah dilakukan *monitoring*, karena terdapat target kelas yang akan dituju dalam analisisnya. Beberapa contoh metode yang dapat digunakan untuk klasifikasi suatu data teks adalah NBC dan SVM.

2.2 Analisis Sentimen

Analisis Sentimen mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan *text mining* yang bertujuan menganalisa pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang yang berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu (Liu, 2011).

Tugas dasar analisis sentimen adalah mengelompokkan teks yang ada dalam sebuah kalimat atau dokumen kemudian menentukan pendapat yang dikemukakan dalam kalimat atau dokumen tersebut apakah bersifat positif, negatif, atau netral (Dehaff, M., 2010). Analisis sentimen juga dapat menyatakan perasaan emosional sedih, gembira, atau marah.

Kita dapat mencari pendapat tentang produk-produk, merek atau orang-orang dan menentukan apakah mereka dilihat positif atau negatif di web (Saraswati, 2011).

Hal ini memungkinkan kita untuk mencari informasi tentang:

- a. Persepsi produk baru.
- b. Persepsi Merek.
- c. Manajemen reputasi.

Ekspresi mengacu pada fokus topik tertentu, pernyataan pada suatu topik mungkin akan berbeda makna dengan pernyataan yang sama pada *subject* yang berbeda. Tujuan dari analisis sentimen adalah untuk menentukan perilaku atau opini dari seorang peneliti dengan memperhatikan suatu topik tertentu. Perilaku bisa mengindikasikan alasan, opini atau penilaian, kondisi kecenderungan (bagaimana si penulis ingin mempengaruhi pembaca).

2.3 Klasifikasi Teks

Klasifikasi teks merupakan proses menemukan pola baru yang belum terungkap sebelumnya. Klasifikasi teks dilakukan dengan memproses dan menganalisa data dalam jumlah besar. Dalam prosesnya, klasifikasi teks melibatkan struktur yang mungkin ter-dapat pada teks dan mengekstraks informasi yang relevan pada teks. Dalam menganalisis sebagian atau keseluruhan teks yang tidak terstruktur, klasifikasi teks mencoba meng-asosiasikan sebagian atau keseluruhan satu bagian teks dengan yang lainnya berdasarkan aturan-aturan tertentu (Miller, 2005). Tantangan dari klasifikasi teks adalah sifat data yang tidak terstruktur dan sulit untuk menangani, sehingga diperlukan proses *text mining*. Diharapkan melalui proses *text mining*, informasi yang ada dapat dikeluarkan secara jelas di dalam teks tersebut dan dapat dipergunakan dalam proses analisis menggunakan alat bantu komputer (Witten dkk, 2011).

2.4 Ketepatan Klasifikasi Model

Tahapan praproses ini dilakukan agar dalam klasifikasi dapat diproses dengan baik. Tahapan dalam praproses teks adalah sebagai berikut:

- a. *Case Folding*, merupakan proses untuk mengubah semua karakter pada teks menjadi huruf kecil. Karakter yang diproses hanya huruf 'a' hingga 'z' dan selain karakter tersebut akan dihilangkan seperti tanda baca titik (.), koma (,), dan angka. (Weiss, 2010)
- b. *Cleansing*, yaitu proses membersihkan *tweet* dari kata yang tidak diperlukan untuk mengurangi *noise*. Kata yang dihilangkan dalam *Twitter* adalah karakter HTML, *emoticons*, *hashtag* (#), *username* (@username), dan *url*.
- c. *Tokenizing*, merupakan proses memecah yang semula berupa kalimat menjadi kata-kata atau memutus urutan string menjadi potongan-potongan seperti kata-kata berdasarkan tiap kata yang menyusunnya. Sehingga dapat dikatakan mengembalikan kata penghubung

- d. *Stopwords*, yakni kosakata yang bukan merupakan kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada kalimat (Dragut, Fang, Sistla, Yu, & Meng, 2009). Kosakata yang dimaksudkan tersebut adalah kata penghubung dan kata keterangan yang bukan merupakan kata unik misalnya “sebuah”, “oleh”, “pada”, dan sebagainya.
- e. *Stemming*, yakni proses untuk mendapatkan kata dasar dengan cara meng-hilangkan awalan, akhiran, sisipan, dan kombinasi dari awalan dan akhiran.

2.5 *Confix-Stripping Stemmer*

Menurut istilah berdasarkan Badan Pusat Statistik (2015), Pada tahun 2007 algoritma *nazief stemmer* kemudian dikembangkan lagi Jelita Asian, dengan menambahkan beberapa perbaikan yang bertujuan untuk meningkatkan hasil *stemming* yang diperoleh. Algoritma ini kemudian dikenal sebagai *confix-stripping stemmer*. Perbaikan tersebut antara lain sebagai berikut:

1. Menggunakan kamus kata dasar yang lebih lengkap.
2. Memodifikasi dan menambahkan aturan pemenggalan untuk tipe awalan yang kompleks.
3. Menambahkan aturan *stemming* untuk kata ulang dan bentuk jamak, misalnya kata (1) 'buku-buku' yang menjadi 'buku', (2) 'berkirim-kiriman' menjadi kirim. Hal ini dilakukan dengan melakukan pemisahan kata tersebut menjadi dua kata yang masing-masing di-*stemming*. Jika *stemming* memberikan kata dasar yang sama, maka keluaran kata dasarnya adalah hasil *stemming* tersebut. Jika hasil *stemming* dua kata tersebut berbeda maka disimpulkan bahwa masukan adalah kata ulang semu dan tidak memiliki bentuk kata dasar lagi.
4. Mengubah urutan *stemming* untuk beberapa kasus tertentu. Algoritma *stemmer* yang diperkenalkan oleh Nazief akan menghilangkan akhiran lebih dahulu, baru diikuti penghilangan awalan. Namun menurut Jelita, cara ini tidak selalu berhasil pada beberapa kata. Sehingga diberikan aturan yang akan mengubah urutan *stemming*, yang mana peng-

hilangan awalan dilakukan terlebih dahulu kemudian diikuti oleh penghilangan akhiran. Aturan ini disebut *rule prece-dence* dan berlaku jika kata memiliki kombinasi awalan-akhiran 'be-lah', 'be-an', 'me-i', 'di-i', 'pe-i', atau 'te-i', misal-nya 'bertaburan', 'melindungi', 'dilengkapi', dan 'teradili'. Berikut merupakan gambar pra proses teks pada data *tweet* hingga membentuk kata kunci.

Gambar 2.1 Contoh hasil *crawling data* dari Twitter API

@mediaindonesia: Tak Diperbolehkan Keluar Ruangan,
Seorang Pendukung AHY-Sylvi Mengamuk
<https://t.co/QvxpNfW6yi> #DebatFinalPilkadaJKT

Dari hasil *crawling data* didapatkan contoh data *tweet*, selanjutnya adalah melakukan *case folding*, yaitu proses mengubah semua karakter teks menjadi huruf kecil, sehingga seperti pada Gambar 2.2 berikut

Gambar 2.2 Contoh hasil *case folding*

@mediaindonesia: tak diperbolehkan keluar ruangan
seorang pendukung ahy sylvi mengamuk
<https://t.co/QvxpNfW6yi> #debatfinalpilkadajkt

Dari hasil *case folding* didapatkan contoh data *tweet*, selanjutnya adalah melakukan *cleansing*, yaitu proses pembersihan *tweet* dari kata yang tidak diperlukan untuk mengurangi *noise*. Sehingga seperti pada Gambar 2.3 berikut

Gambar 2.3 Contoh hasil *cleansing data*

tak diperbolehkan keluar ruangan seorang pendukung ahy
sylvi mengamuk

Dari hasil *cleansing* didapatkan contoh data *tweet*, selanjutnya adalah melakukan *stemming*, yaitu kata-kata yang terda-

pat pada *tweet* akan dibandingkan oleh pemisahan kata hubung didapatkan kata kunci. Sehingga seperti pada Gambar 2.4 berikut

Gambar 2.4 Contoh hasil *steaming*

tak boleh keluar ruangan seorang dukung ahy sylvi amuk

Selanjutnya adalah menentukan bobot dari kata kunci analisis sentimen pada *tweet* tersebut seperti pada tabel 2.1

Tabel 2.1 Ilustrasi hasil pembobotan kata kunci

<i>Tweet</i>	tak	boleh	keluar	dukung	amuk	Strategi	Kelas
1	0	1	0	1	0	0	Positif
2	1	0	1	0	1	0	Negatif
3	0	0	0	1	0	1	Positif
4	0	0	0	1	0	0	Positif
5	1	0	0	0	0	1	Negatif
6	0	1	0	0	0	0	Positif
7	1	0	1	1	0	0	Negatif
8	0	0	1	0	0	1	Negatif
9	0	0	0	0	0	1	Positif
10	0	0	1	1	0	0	Negatif
11	0	1	0	1	0	1	Positif
12	0	0	0	1	0	1	Positif
13	1	0	1	0	0	0	Negatif
Total	4	3	5	7	1	6	

Dari hasil pembobot tersebut didapatkan kata kunci dari salah satu *tweet* selanjutnya, untuk menentukan kata kunci selanjutnya akan dihitung frekuensi kumulatif dari kata kunci yang sudah ada, jika belum ada maka akan dijadikan kata kunci yang baru.

2.6 *Naive Bayes Classifier*

Teorema Bayes merupakan teorema yang mengacu konsep probabilitas bersyarat (Tan et al, 2006). Teorema Bayes dapat dinotasikan pada persamaan berikut:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (2.1)$$

Metode *naive bayes* yang sering disebut sebagai *naive bayes classification* (NBC), merupakan salah satu metode yang dapat mengklasifikasikan teks. Kelebihan NBC adalah sederhana tetapi memiliki akurasi yang tinggi. Dalam algoritma NBC setiap dokumen direpresentasikan dengan pasangan atribut “ $a_1, a_2, a_3, \dots, a_n$ ” dimana a_1 adalah kata pertama, a_2 adalah kata kedua dan seterusnya. Sedangkan V adalah himpunan kategori berita. Pada saat klasifikasi algoritma akan mencari probabilitas tertinggi dari semua kategori dokumen yang diujikan (V_{MAP}). Adapun persamaan V_{MAP} adalah sebagai berikut:

$$V_{MAP} = \operatorname{argmax}_{v_j \in V} P(v_j | a_1, a_2, \dots, a_n) \quad (2.2)$$

Dengan menggunakan teorema Bayes, maka persamaan (2.2) dapat ditulis menjadi,

$$V_{MAP} = \operatorname{argmax}_{v_j \in V} \frac{P(a_1, a_2, \dots, a_n | v_j) P(v_j)}{P(a_1, a_2, \dots, a_n)} \quad (2.3)$$

Karena nilai $P(a_1, a_2, a_3, \dots, a_n)$ untuk semua v_j besarnya sama maka nilainya dapat diabaikan, sehingga persamaan (2.3) menjadi:

$$V_{MAP} = \operatorname{argmax}_{v_j \in V} P(a_1, a_2, \dots, a_n | v_j) P(v_j) \quad (2.4)$$

Naive bayes classifier menyederhanakan hal ini dengan mengasumsikan bahwa didalam setiap kategori, setiap atribut bebas bersyarat satu sama lain (Tan et al, 2006). NBC didapatkan dari frekuensi kata-kata setiap *tweet* yang di akumulasi berdasarkan kata terbanyak dari *tweet* yang sudah ditentukan. Dengan kata lain persamaan (2.4) dapat dituliskan sebagai berikut:

$$P(a_1, a_2, \dots, a_n | v_j) = \prod_i P(a_i | v_j) \quad (2.5)$$

Kemudian apabila persamaan (2.5) disubstitusikan ke persamaan (2.4), maka akan menghasilkan persamaan:

$$V_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j) \quad (2.6)$$

Nilai $P(v_j)$ dihitung pada saat data *training*, didapat dengan rumus sebagai berikut:

$$P(v_j) = \frac{|doc\ j|}{|training|} \quad (2.7)$$

dimana $|doc\ j|$ merupakan jumlah dokumen (artikel berita) yang memiliki kategori j dalam *training*. Sedangkan $|training|$ merupakan jumlah dokumen (artikel berita) dalam contoh yang digunakan untuk *training*. Untuk probabilitas kata a_i untuk setiap kategori $P(a_i | v_j)$, dihitung pada saat *training*. Dimana,

$$P(a_i | v_j) = \frac{|n_i + 1|}{|n + kosakata|} \quad (2.8)$$

dimana n_i adalah jumlah kemunculan kata a_i dalam dokumen yang berkategori v_j , sedangkan n adalah banyaknya seluruh kata dalam dokumen dengan kategori v_j dan $|kosakata|$ adalah banyak-nya kata dalam contoh pelatihan. Dari ilustrasi kata kunci yang terbentuk pada tabel 2.1 maka langkah pertama adalah menentukan *tweet* yang di ambil secara acak oleh data *tweet* sebanyak 13 *tweet*, kemudian memisah antara sentimen yang positif dan negatif berdasarkan yang sudah ditentukan oleh pe-neliti secara manual dari persepsi yang ditentukan oleh peneliti dengan survey langsung kepada responden. Data NBC merupakan data yang bertuju pada jumlah frekuensi pada kata didalam *tweet* dari ketiga pasangan calon tersebut.

Tabel 2.2 Ilustrasi perhitungan NBC pada sentimen positif

<i>Tweet</i>	Tak	Boleh	Keluar	dukung	amuk	strategi	Kelas
1	0	1	0	1	0	0	+
2	0	0	0	1	0	1	+
3	0	0	0	1	0	0	+
4	0	1	0	0	0	0	+
5	0	0	0	0	0	1	+
6	0	1	0	1	0	1	+
7	0	0	0	1	0	1	+
Total	0	2	0	4	0	4	

Menghitung dari probabilitas kata kunci yang terdapat pada sentimen positif didapatkan

$$p(+) = \frac{7}{13} = 0,538; \quad n(+) = 10;$$

Selanjutnya mendapatkan probabilitas kata kunci di setiap kategori kelas didapatkan

Tabel 2.3 Ilustrasi probabilitas NBC pada sentimen positif

Kata kunci	Probabilitas
Boleh	$P(a_i v_j) = \frac{2+1}{10+6} = 0,1875$
Dukung	$P(a_i v_j) = \frac{4+1}{10+6} = 0,3125$
strategi	$P(a_i v_j) = \frac{4+1}{30+6} = 0,3125$

Selanjutnya untuk sentimen negatif seperti berikut

Tabel 2.4 Ilustrasi perhitungan NBC pada sentimen negatif

<i>Tweet</i>	tak	boleh	Keluar	dukung	amuk	strategi	Kelas
1	1	0	1	0	1	0	+
2	1	0	1	0	0	0	+
3	1	0	1	0	0	0	+
4	0	0	1	0	0	0	+
5	1	0	0	0	0	0	+
6	0	0	1	0	0	0	+
Total	4	0	5	0	1	0	

Menghitung dari probabilitas kata kunci yang terdapat pada sentimen positif didapatkan

$$p(-) = \frac{6}{13} = 0,362; \quad n(-) = 9;$$

Selanjutnya mendapatkan probabilitas kata kunci di setiap kategori kelas didapatkan

Tabel 2.5 Ilustrasi probabilitas NBC pada sentimen positif

Kata kunci	Probabilitas
Tak	$P(a_i v_j) = \frac{4+1}{9+6} = 0,3333$
Amuk	$P(a_i v_j) = \frac{1+1}{9+6} = 0,1333$
Dukung	$P(a_i v_j) = \frac{1+1}{9+6} = 0,1333$

Pada data uji, dilakukan perhitungan V_{MAP} untuk menentukan probabilitas tertinggi dari masing-masing kelas berdasarkan dari proses pelatihan. Nilai probabilitas tertinggi merupakan sentimen dari data *tweet* tersebut.

2.7 Term Frequency Inverse Document Frequency

Term Frequency Inverse Document Frequency (TF-IDF) merupakan pembobot yang dilakukan setelah ekstraksi artikel berita (Ariadi, & Fithriasari, 2015). Proses metode TF-IDF adalah menghitung bobot dengan cara integrasi antara *term frequency* (*tf*) dan *inverse document frequency* (*idf*). Langkah dalam TF-IDF adalah untuk menemukan jumlah kata yang kita ketahui (*tf*) setelah dikalikan dengan berapa banyak artikel berita dimana suatu kata itu muncul (*idf*). Rumus dalam menumakan pembobot dengan TF-IDF adalah sebagai berikut :

$$w_{ij} = tf_{ij} \times idf, \quad idf = \log \left(\frac{N}{df_j} \right) \quad (2.9)$$

dimana w_{ij} adalah bobot dari kata i pada artikel ke j , N merupakan jumlah seluruh dokumen, tf_{ij} adalah jumlah kemunculan kata i pada dokumen j , df_j adalah jumlah artikel j yang me-ngandung kata i . TF-IDF dilakukan agar data dapat dianalisis menggunakan *support*

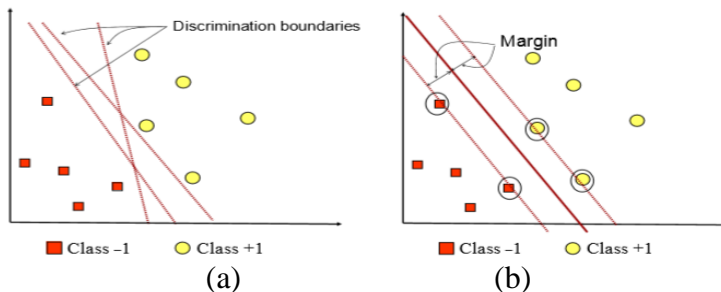
vector machine. *Support vector machine* akan dijelaskan lebih jelas pada sub bab 2.8.

2.8 Support Vector Machine

Support Vector Machine (SVM) adalah salah satu metode statistika mutakhir yang sedang berkembang dengan pesat. SVM pertama kali diperkenalkan oleh Vapnik tahun 1992 di *Annual Workshop on Computational Learning Theory*. SVM adalah salah satu dari beberapa metode yang dikembangkan untuk mengatasi permasalahan yang tidak bisa diselesaikan dengan metode statistika klasik, terutama pada kasus klasifikasi dan prediksi. SVM dikembangkan dengan prinsip *linier classifier*. Namun dalam kasus nyata sering dijumpai data yang tidak linier sehingga dikembangkan SVM untuk kasus nonlinier dengan memasukkan konsep kernel.

2.8.1 Support Vector Machine Linier

Support Vector Machine (SVM) Konsep dari SVM adalah berusaha menemukan *hyperplane* yang optimum pada *input space*. Fungsi dari *hyperplane* itu digunakan sebagai pemisah dua buah kelas pada *input space*. Kelas sering disimbolkan dengan -1 dan +1. Pada Gambar 2.5 di ilustrasikan *hyperplane* pada metode SVM. *Pattern* pada kelas -1 ditandai dengan warna merah. Sedangkan *pattern* pada kelas +1, ditandai dengan warna kuning (lingkaran).



Gambar 2.5 Ilustrasi Hyperplane pada Metode SVM

Gambar 2.5.(a) menunjukkan alternatif garis pemisah antara dua kelas (*discriminant boundaries*). Garis pemisah yang terbaik adalah yang memiliki *margin hyperplane* maksimum. *Margin* adalah jarak antara *hyperplane* dengan *pattern* terdekat pada masing-masing kelas. *Pattern* yang paling dekat disebut sebagai *support vector*. Pada Gambar 2.5.(b), *pattern* yang dilingkari adalah *support vector* untuk tiap kelas. Sedangkan, garis tebal dalam Gambar 2.5.(b) adalah *hyperplane* terbaik karena berada di tengah-tengah kedua kelas. Proses mencari letak *hyperplane* ini adalah inti dari metode SVM.

2.8.2 Support Vector Classification

Data yang tersedia akan dinotasikan dengan $\vec{x}_i \in \Re^d$. Sedangkan untuk label masing - masing diberi dan di notasikan sebagai $y_i \in \{-1, +1\}, i = 1, 2, \dots, l$, dimana l adalah banyaknya data yang digunakan. Misalkan diketahui X memiliki pola dimana \mathbf{x}_i termasuk ke dalam kelas maka \mathbf{x}_i diberi label (target) $y_i = +1$ dan $y_i = -1$. Asumsi yang digunakan ada kelas -1 dan +1 dapat terpisah secara sempurna oleh *hyperplane* berdimensi d , yang didefinisikan sebagai berikut.

$$\vec{w} \cdot \vec{x} + b = 0.$$

Pattern \mathbf{X}_i masuk ke dalam kelas -1 (sampel negatif) jika memenuhi pertidaksamaan berikut:

$$\vec{w} \cdot \vec{x} + b \leq -1.$$

Sedangkan *pattern* \mathbf{X}_i masuk ke dalam kelas +1 (sampel positif) jika memenuhi pertidaksamaan berikut:

$$\vec{w} \cdot \vec{x} + b \geq +1.$$

Margin terbesar bisa diperoleh dengan memaksimumkan nilai antara *hyperplane* dengan titik terdekatnya, yaitu $1/\|\vec{w}\|$. Hal ini sebagai *Quadratic Programming (QP) problem*, yaitu mencari titik minimal persamaan (2.10) dengan batasan persamaan (2.11)

$$\min_{\vec{w}} \tau(\vec{w}) = \frac{1}{2} \|\vec{w}\|^2 \quad (2.10)$$

$$y_i [\vec{w} \cdot \vec{x}_i + b] \geq 0, i = 1, 2, \dots, l. \quad (2.11)$$

Problem ini dapat diatasi dengan menggunakan teknik komputasi, diantaranya adalah Lagrange Multiplier.

$$L(\vec{w}, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^l \alpha_i \{y_i [\vec{w} \cdot \vec{x}_i + b] - 1\} \quad (2.12)$$

dengan $i=1, 2, \dots, l$.

α_i adalah *Lagrange Multiplier* yang nilainya nol atau positif ($\alpha_i \geq 0$). Persamaan (2.11) dapat dioptimalkan dengan meminimalkan L terhadap \vec{w} dan b , dan memaksimalkan L terhadap α_i dengan menggunakan sifat bahwa pada titik optimum gradien $L = 0$, persamaan (2.11) dapat ditulis sebagai berikut.

Memaksimumkan:

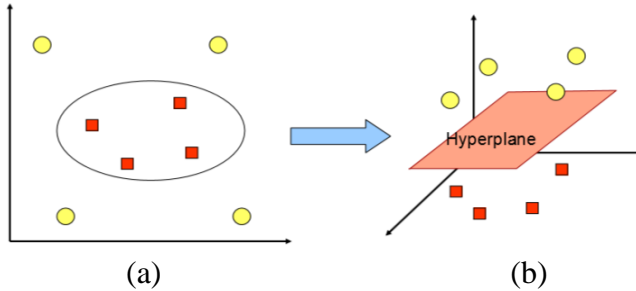
$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j, \quad (2.13)$$

dimana, $\alpha_i \geq 0$ ($i=1, 2, \dots, l$) $\sum_{i=1}^l \alpha_i y_i = 0$.

Berdasarkan persamaan (2.13) akan diperoleh α_i yang kebanyakan bernilai positif. Data yang berkorelasi dengan α_i yang positif inilah yang disebut sebagai *support vector*.

2.8.3 Fungsi Kernel pada SVM

Pada kasus nyata, sangat jarang dijumpai masalah yang bersifat *linier separable*. Sebagai metode yang dikembangkan untuk kasus linier, SVM membutuhkan sebuah fungsi yang mampu membuat pemisah yang tidak linier. Fungsi yang sering digunakan untuk mengatasi hal tersebut adalah fungsi kernel. Pada kasus nonlinier SVM, data \vec{x} terlebih dahulu dipetakan oleh fungsi $\Phi(\vec{x})$ ke dalam ruang vektor yang berdimensi lebih tinggi. Setelah mendapat ruang vektor yang baru, kemudian *hyperplane* bisa dikonstruksikan untuk memisahkan kedua kelas sentimen yaitu positif maupun sentimen negatif didalam *tweet* yang sudah ditentukan secara manual.



Gambar 2.6 Fungsi Φ memetakan data ke ruang vektor

Gambar 2.6.(a) mengilustrasikan contoh kasus data yang berada di dimensi dua dan tidak dapat dipisahkan dengan menggunakan *hyperplane* yang linier. Kemudian Gambar 2.6.(b) adalah pemetaan data pada *input space* ke dimensi yang lebih tinggi (dimensi tiga) melalui fungsi Φ . Setelah dipetakan ke dimensi yang lebih tinggi kelas merah dan kelas kuning dapat dipisahkan dengan *hyperplane*. Notasi dari pemetaan ini ditunjukkan oleh persamaan berikut:

$$\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^q ; d < q.$$

Pemetaan yang dilakukan tidak mengubah karakteristik atau dalam kata lain tetap menjaga tipologi data. Artinya, dua data yang berjarak dekat pada *input space* akan tetap berjarak dekat juga pada *feature space*. Sebaliknya, dua data yang berjarak jauh pada *input space* akan tetap berjarak jauh pada *feature space*.

Setelah melakukan pemetaan, langkah selanjutnya dalam proses SVM adalah menemukan titik-titik *support vector*. Untuk menemukan titik-titik *support vector*, digunakan *dot product* dari data yang sudah ditransformasi pada ruang yang berdimensi lebih tinggi, yaitu $\Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$. Akan tetapi, transformasi Φ tidak dapat diketahui dan sangat sulit dipahami sehingga perhitungan *dot product* tersebut dapat diganti dengan fungsi kernel $K(\vec{x}_i, \vec{x}_j)$ yang mendefinisikan secara implisit transformasi Φ . Gunn tahun 1998 merumuskan fungsi kernel sebagai berikut:

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j).$$

Dengan menggunakan fungsi kernel proses menentukan *support vector* menjadi lebih mudah karena cukup dengan mengetahui fungsi kernel yang dipakai, tanpa perlu mengetahui wujud dari fungsi nonlinier Φ . Beberapa fungsi kernel yang biasa digunakan ditunjukkan pada Tabel 2.6.

Tabel 2.6 Fungsi Kernel yang umum pada SVM

Jenis Kernel	Fungsi
Polynomial	$K(\vec{x}_i, \vec{x}_j) = ((\vec{x}_i, \vec{x}_j) + 1)^p$ dimana $p=1, \dots$
Gaussian Radial Basis Function (RBF)	$K(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\ \vec{x}_i - \vec{x}_j\ ^2}{2\sigma^2}\right)$
Sigmoid	$K(\vec{x}_i, \vec{x}_j) = \tanh(\alpha \cdot \vec{x}_i \cdot \vec{x}_j + \beta)$

Selanjutnya hasil klasifikasi dari data \vec{x} diperoleh dari persamaan berikut :

$$\begin{aligned}
 f[\Phi(\vec{x})] &= \vec{w} \Phi(\vec{x}) + \vec{b} \\
 f[\Phi(\vec{x})] &= \sum_{i=1, SVs} \alpha_i y_i \Phi(\vec{x}) \cdot \Phi(\vec{x}_i) + \vec{b} \\
 &= \sum_{i=1, SVs} \alpha_i y_i K(\vec{x}_i, \vec{x}_j) + \vec{b} .
 \end{aligned}$$

Support vector pada persamaan di atas adalah subset dari training set yang terpilih sebagai *support vector*, dengan kata lain data \vec{x}_i yang berkorespondensi pada $\alpha_i \geq 0$. Dalam ilustrasinya sebagai berikut; $\vec{w} \cdot \vec{x} + b \geq +1$ untuk semua *tweet* yang mengandung makna positif dan $\vec{w} \cdot \vec{x} + b \geq -1$ semua *tweet* yang mengandung makna negatif. Pembahasan pada SVM ini akan sama dengan pembahasan pada NBC, perbedaannya adalah ditambahkan parameter untuk SVM dengan kernel RBF yaitu C dan γ , kernel polynomial dengan parameter C , γ , dan p , sedangkan untuk kernel linier tidak ditambahkan parameter. Saat menentukan kombinasi parameter C dan γ , terlebih dahulu harus dideskripsikan *range* dari kedua parameter untuk dikombinasikan. Parameter C yang paling tepat untuk SVM adalah antara 10^{-2} hingga 10^4 (Huang et al.,

2007). Pada penelitian ini juga di-gunakan *range* 10^{-2} hingga 10^4 untuk parameter C . Kemudian untuk parameter γ , Huang dkk tahun 2007 merekomendasikan nilai antara $\frac{10^{-3}}{\rho}$ hingga $\frac{1,9}{\rho}$ dengan asumsi nilai ρ adalah 0,5 maka diperoleh nilai antara 0,02 hingga 3,8. Berikut merupakan ilustrasi dari perhitungan *Support Vector Machine* dari data *tweet*.

Tabel 2.7 Ilustrasi *tweet* yang umum pada SVM

No	<i>Tweet</i>	Sentimen
1	sadis banget closing statement singkat tegas tusuk paslon dan jelas	Positif
2	ini bukan tentang kita para cagub dan cawagub terus dia habis durasi ngomong soal dia	Negatif
3	pasang calon nomor dan terkadang sesat	Positif
4	baik paslon no1 atau pun paslon no yg menang dan pimpin dki alhamdulillah yang penting bukan	Negatif
5	keren pake banget ini closing statement nya bener pukul yg keras buat sama	Positif

Pada tabel 2.7 ilustrasi dari *tweet* yang digunakan dalam perhitungan svm akan di tentukan *td-idf* (jumlah bobot keseluruhan per-kata) dari kelima *tweet* tersebut.

Tabel 2.8 Ilustrasi *td-idf* (jumlah bobot keseluruhan per-kata)

Term	TF		IDF	
	Teks 1 (Q)	Teks 2 (D1)	Df	log(n/df)
sadis	1	1	2	0
b banget	2	0	1	0.30103
c closing	1	0	1	0.30103
...
s sama	1	0	1	0.30103

Didapatkan tabel 2.8 ilustrasi jumlah bobot keseluruhan perkata yang di gunakan dalam *tweet* untuk perhitungan SVM, selanjutnya adalah menentukan nilai *support vector* dari masing-masing *tweet* yang di gunakan kedalam *hyperplane*.

Tabel 2.9 Ilustrasi *tweet* dalam menentukan *support vector*

No	<i>Tweet</i>	Perhitungan	Sentimen
1	sadis banget closing statement singkat tegas tusuk paslon dan jelas	$0 + 0.30103 +$ $0.30103 + \dots +$ $0.30103 = 2.40824$	Positif
2	ini bukan tentang kita para cagub dan cawagub terus dia habis durasi ngomong soal dia	$0 + 0 + 0 + 0 +$ $0.30103 + \dots + 0 =$ 1.50515	Negatif
3	pasang calon nomor dan terkadang sesat	$0.30103 + 0.30103 +$ $0.30103 + \dots +$ $0.30103 = 1.50515$	Positif
4	baik paslon no atau pun paslon no yg menang dan pimpin dki alhamdulillah yang penting bukan	$0 + 0.30103 + 0 + 0$ $+ 0 + \dots + 0 =$ 2.10721	Negatif
5	keren pake banget ini closing statement nya bener pukul yg keras buat sama	$0.30103 + 0 + 0 + 0$ $+ \dots + 0 = 1.80618$	Positif

2.9 Pengukuran Performa Klasifikasi

Pengukuran performa dilakukan untuk melihat hasil yang didapatkan dari klasifikasi. Terdapat beberapa cara untuk mengukur performa, beberapa cara yang sering digunakan adalah dengan menghitung akurasi, *recall*, dan *precision*. (Hotho, Nurnberger, & Paass, 2005). Akurasi merupakan persentase dari total dokumen yang teridentifikasi secara tepat dalam proses klasifikasi. *Recall* mengindikasikan sebagian kecil dari dokumen yang relevan diambil. *Precision* mengkuantifikasi fraksi dokumen diambil yang sebenarnya relevan, dalam contoh milik kelas sasaran. Masing-masing persamaannya adalah sebagai berikut :

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.14)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2.15)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.16)$$

TP adalah *True Positive*, FP adalah *False Positive*, TN adalah *True Negative*, dan FN adalah *False Negative*.

F-measure merupakan kompromi dari recall dan precision untuk mengukur kinerja keseluruhan pengklasifikasi. Berikut merupakan cara perhitungan *f-measure*. (Hotho dkk, 2005)

$$F = \frac{2 \times recall \times precision}{recall + precision} \quad (2.17)$$

2.10 Twitter

Twitter adalah situs web dimiliki dan dioperasikan oleh *Twitter, Inc.*, yang menawarkan jaringan sosial berupa *microblog*. Disebut *microblog* karena situs ini memungkinkan penggunaanya mengirim dan membaca pesan blog seperti pada umumnya namun terbatas hanya sejumlah 140 karakter yang ditampilkan pada halaman profil pengguna. *Twitter* memiliki karakteristik dan format penulisan yang unik dengan simbol ataupun aturan khusus. Pesan dalam *Twitter* dikenal dengan sebutan *tweet*. Kelebihan *Twitter* dibandingkan dengan sosial media lainnya ialah mempunyai fitur *trending topic*, yaitu tema bahasan yang sedang menjadi trend di *Twitter* dan biasanya disertai dengan penggunaan *hashtag* (#). *Trending topic* biasanya berhubungan dengan suatu tema yang sedang hangat dibicarakan oleh para pengguna *Twitter*.

2.11 Pemilihan Kepala Daerah (Pilkada) DKI Jakarta 2017

Pemilihan Kepala Daerah baik tingkat provinsi maupun tingkat kabupaten dan kota dalam lingkup wilayah atau kawasan tertentu yang dilakukan secara serentak atau dalam waktu yang bersamaan. Tujuan dilaksanakannya pilkada serentak adalah untuk efektifitas dan efisiensi dalam pelaksanaannya, dengan harapan dapat dilakukannya penghematan waktu, energi, dan anggaran pilkada. Pilkada serentak dilakukan pertama kali dilakukan pada daerah yang masa jabatan pemerintahannya berakhir di tahun 2015.

Selanjutnya ditahun 2017, pilkada serentak tahun 2017 diikuti oleh 7 provinsi, 76 kabupaten, dan 18 kota, salah satunya DKI Jakarta. Ada 3 calon pasangan yang maju di putaran pertama bertarung untuk menjadi orang nomer 1 di Ibukota yaitu pasangan Agus Harimurti Yudhoyono – Sylviana Murni, Basuki Tjahaja Purnama – Syaiful Djarot, dan Anies Baswedan Rasyid dan Sandiaga Salahudin Uno. Dari hasil penghitungan suara KPU ditetapkan perolehan suara pasangan calon nomor urut 1 sebesar 937.955 atau 17,05 persen. Selanjutnya pasangan calon nomor urut 2 sebesar 2.364.577 atau 42,99%. Terakhir, pasangan calon nomor urut 3 meraup 2.197.333 suara atau 39,95 persen. Iki dan dioperasikan oleh *Twitter, Inc.*, yang menawarkan jaringan sosial berupa *microblog*.

(halaman ini sengaja dikosongkan)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data Sumber data yang akan digunakan dalam penelitian ini adalah *tweet* dari pengguna *Twitter* di Indonesia dengan *keywords* dari masing-masing calon Gubernur DKI Jakarta, yaitu mengambil data *tweet* masing-masing berjumlah 1000. Data yang digunakan dalam penelitian ini baik untuk tahapan awal adalah data yang terkumpul dari hasil crawling 3 akun twitter yaitu @AgusYudhoyono, @basuki_btp, dan @aniesbaswedan serta hubungan dari antar calon maupun ketiganya. Data tersebut diambil dari periode masa tenang putaran 1 Pilkada Serentak dari 11 Februari 2017 – 14 Februari 2017. Pemilihan waktu pada masa tenang diakrenakan pengguna *Twitter* sudah mempunyai pilihannya setelah masa kampanye dan debat panelis terbuka yang telah dilakukan sebanyak 3 kali.

3.2 Struktur Data

Struktur data yang diambil dari *website www.twitter.com* dengan bantuan Twiter API *software R 3.3.1* dibuat seperti pada Tabel 3.1.

Tabel 3.1 Struktur Data Sebelum *Preprocessing*

No.	Nama Akun	<i>Tweet (y)</i>	Klasifikasi Sentimen
1	@AgusYudhoyono	y_1	isi_1
2		y_2	isi_2
⋮		⋮	⋮
⋮		⋮	⋮
1000		y_{1000}	isi_{1000}
1	@basuki_btp	y_1	isi_1
2		y_2	isi_2
⋮		⋮	⋮
⋮		⋮	⋮
1000		y_{1000}	isi_{1000}

Tabel 3.1 Struktur Data Sebelum *Preprocessing* (lanjutan)

No.	Nama Akun	Tweet (y)	Klasifikasi Sentimen
1		y ₁	isi ₁
2		y ₂	isi ₂
⋮	@AniesBaswedan	⋮	⋮
⋮		⋮	⋮
⋮		⋮	⋮
1000		y ₁₀₀₀	isi ₁₀₀₀
1		y ₁	isi ₁
2		y ₂	isi ₂
⋮	@AgusYudhoyono	⋮	⋮
⋮	& @basuki_btp	⋮	⋮
⋮		⋮	⋮
n		y _n	Isi _n
1		y ₁	isi ₁
2	@AgusYudhoyono	y ₂	isi ₂
⋮	&	⋮	⋮
⋮	@AniesBaswedan	⋮	⋮
⋮		⋮	⋮
n		y _n	Isi _n
1		y ₁	isi ₁
2		y ₂	isi ₂
⋮	@basuki_btp &	⋮	⋮
⋮	@AniesBaswedan	⋮	⋮
⋮		⋮	⋮
n		y _n	Isi _n
1		y ₁	isi ₁
2	@AgusYudhoyono,	y ₂	isi ₂
⋮	@basuki_btp, &	⋮	⋮
⋮	@AniesBaswedan	⋮	⋮
⋮		⋮	⋮
n		y _n	Isi _n

Struktur data pada Tabel 3.1 menunjukkan jumlah data dan isi menunjukkan klasifikasi sentimen bahwa *tweet* tersebut mengandung sentiment positif atau negatif. Setelah semua data didapatkan, tahap selanjutnya adalah melakukan *pre-processing*. Struktur data hasil *pre-processing* disajikan pada Tabel 3.2.

Pada Tabel 3.2, notasi I menunjukkan jumlah kata kunci yang terbentuk dari ketiga calon Gubernur DKI Jakarta, menunjukkan jumlah kata kunci berjalan hingga (w_1) yaitu kata kunci yang didapatkan dari hasil data *tweet* yang telah ditentukan. Sementara untuk hubungan antar ketiga calon gubernur DKI Jakarta menggunakan kata kunci yang telah ditentukan.

Tabel 3.2 Struktur Data Setelah *Pre-processing*

No.	Nama Akun	<i>Tweet</i> (y)	Klasifikasi Sentimen	Kata Kunci (w_1)	Kata Kunci (w_2)	...	Kata Kunci (w_1)
1	@AgusYudhoyono	y_1	isi_1	$w1,1,1$	$w1,1,2$...	$w1,1,I$
2		y_2	isi_2	$w2,1,1$	$w2,1,2$...	$w2,1,I$
.	
.	
1000		y_{1000}	isi_{1000}	$w1000;1;1$	$w1000;1;2$...	$w1000;1;I$
1	@basuki_btp	y_1	isi_1	$w1,1,1$	$w1,1,2$...	$w1,1,I$
2		y_2	isi_2	$w2,1,1$	$w2,1,2$...	$w2,1,I$
.	
.	
1000		y_{1000}	isi_{1000}	$w1000;1;1$	$w1000;1;2$...	$w1000;1;I$
1	@AniesBaswedan	y_1	isi_1	$w1,1,1$	$w1,1,2$...	$w1,1,I$
2		y_2	isi_2	$w2,1,1$	$w2,1,2$...	$w2,1,I$
.	
.	
1000		y_{1000}	isi_{1000}	$w1000;1;1$	$w1000;1;2$...	$w1000;1;I$
1	@AgusYudhoyono & @basuki_btp	y_1	isi_1	$w1,1,1$	$w1,1,2$...	$w1,1,I$
2		y_2	isi_2	$w2,1,1$	$w2,1,2$...	$w2,1,I$
.	
.	
n		y_n	isi_{1000}	$w1000;1;1$	$w1000;1;2$...	$w1000;1;I$

Tabel 3.2 Struktur Data Setelah *Pre-processing* (lanjutan)

No.	Nama Akun	<i>Tweet</i> (<i>y</i>)	Klasifikasi Sentimen	Kata Kunci (<i>w</i> ₁)	Kata Kunci (<i>w</i> ₂)	...	Kata Kunci (<i>w</i> _I)
1	@AgusYudhoyono	<i>y</i> ₁	isi ₁	<i>w</i> _{1,1,1}	<i>w</i> _{1,1,2}	...	<i>w</i> _{1,1,I}
2		<i>y</i> ₂	isi ₂	<i>w</i> _{2,1,1}	<i>w</i> _{2,1,2}	...	<i>w</i> _{2,1,I}
⋮		⋮	⋮	⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮	⋮	⋮
1000		<i>y</i> ₁₀₀₀	isi ₁₀₀₀	<i>w</i> _{1000;1;1}	<i>w</i> _{1000;1;2}	...	<i>w</i> _{1000;1;I}
1	@basuki_btp	<i>y</i> ₁	isi ₁	<i>w</i> _{1,1,1}	<i>w</i> _{1,1,2}	...	<i>w</i> _{1,1,I}
2		<i>y</i> ₂	isi ₂	<i>w</i> _{2,1,1}	<i>w</i> _{2,1,2}	...	<i>w</i> _{2,1,I}
⋮		⋮	⋮	⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮	⋮	⋮
1000		<i>y</i> ₁₀₀₀	isi ₁₀₀₀	<i>w</i> _{1000;1;1}	<i>w</i> _{1000;1;2}	...	<i>w</i> _{1000;1;I}
1	@AniesBaswedan	<i>y</i> ₁	isi ₁	<i>w</i> _{1,1,1}	<i>w</i> _{1,1,2}	...	<i>w</i> _{1,1,I}
2		<i>y</i> ₂	isi ₂	<i>w</i> _{2,1,1}	<i>w</i> _{2,1,2}	...	<i>w</i> _{2,1,I}
⋮		⋮	⋮	⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮	⋮	⋮
1000		<i>y</i> ₁₀₀₀	isi ₁₀₀₀	<i>w</i> _{1000;1;1}	<i>w</i> _{1000;1;2}	...	<i>w</i> _{1000;1;I}
1	@AgusYudhoyono & @basuki_btp	<i>y</i> ₁	isi ₁	<i>w</i> _{1,1,1}	<i>w</i> _{1,1,2}	...	<i>w</i> _{1,1,I}
2		<i>y</i> ₂	isi ₂	<i>w</i> _{2,1,1}	<i>w</i> _{2,1,2}	...	<i>w</i> _{2,1,I}
⋮		⋮	⋮	⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮	⋮	⋮
n		<i>y</i> _n	isi ₁₀₀₀	<i>w</i> _{1000;1;1}	<i>w</i> _{1000;1;2}	...	<i>w</i> _{1000;1;I}

3.3 Langkah Analisis

Pada penelitian ini terdiri atas 2 tahapan proses yaitu tahap *pre-processing* (*testing*) dan tahap *testing*. Pada tahap *pre-processing* proses-proses yang dilakukan yaitu tahap klasifikasi sentimen terhadap *tweet* yang sudah diketahui mengandung positif atau negatif. Tujuan dari tahap *pre-processing* adalah untuk

mencari *keyword* beserta probabilitasnya yang nantinya akan digunakan pada proses *testing*. Adapun langkah-langkahnya adalah sebagai berikut.

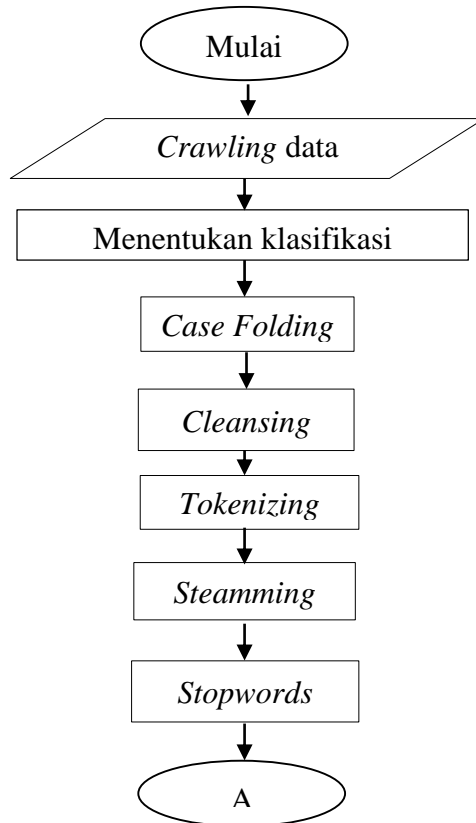
1. Menyiapkan data *tweet*, daftar *stopwords*, dan kata dasar.
 - a) Data *tweet* dalam penelitian ini diperoleh dengan *crawling* data di *twitter* menggunakan program R.
 - b) Penentuan klasifikasi sentimen pada *tweet*.
 - c) Daftar *stopwords*, didapatkan pada tesis F. Tala yang berjudul “*A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia*”.
 - d) Kata dasar ini diambil dari kamus besar bahasa Indonesia.
 - e) Pengujian dilakukan menggunakan validasi silang (*cross validation*) sebanyak 10 kali (10 *folds cross validation*), yaitu dengan membagi data uji menjadi 10 *sub samples*, Untuk rasio data uji dimulai dari 10%, naik 10% setiap kali uji sampai dengan 90%.
2. Praproses Teks
 - a) Melakukan *case folding*, proses untuk mengubah semua karakter pada teks menjadi huruf kecil.
 - b) Selanjutnya *cleansing*, proses pembersihan *tweet* dari kata yang tidak diperlukan untuk mengurangi *noise*. Kata yang dihilangkan dalam *Twitter* adalah karakter HTML, *emoticons*, *hashtag*(#), *username* (@*username*), dan *url*.
 - c) Kemudian melakukan proses *tokenizing* untuk memecah kalimat menjadi kata per kata.
 - d) Melakukan *stemming* pada kata-kata yang tersisa pada dokumen teks untuk mendapatkan kata dasar. Pada tahap ini dilakukan algoritma *confix-stripping stemmer* untuk mendapatkan kata dasar. untuk membandingkan hasil pemenggalan kata dengan kata dasar maka proses ini membutuhkan kamus kata dasar yang telah disiapkan sebelumnya.
 - e) Kemudian dilakukan proses *stopping* berdasarkan *stoplist* yang berisi *stopwords* yang telah ditentukan

sebelumnya. Kata-kata yang terdapat pada artikel berita akan dibandingkan dengan daftar *stopwords*, jika terdapat kata-kata yang terdapat pada *stopwords* maka kata tersebut akan dihapus dari *tweet* sehingga ditemukan kata kunci yang identik.

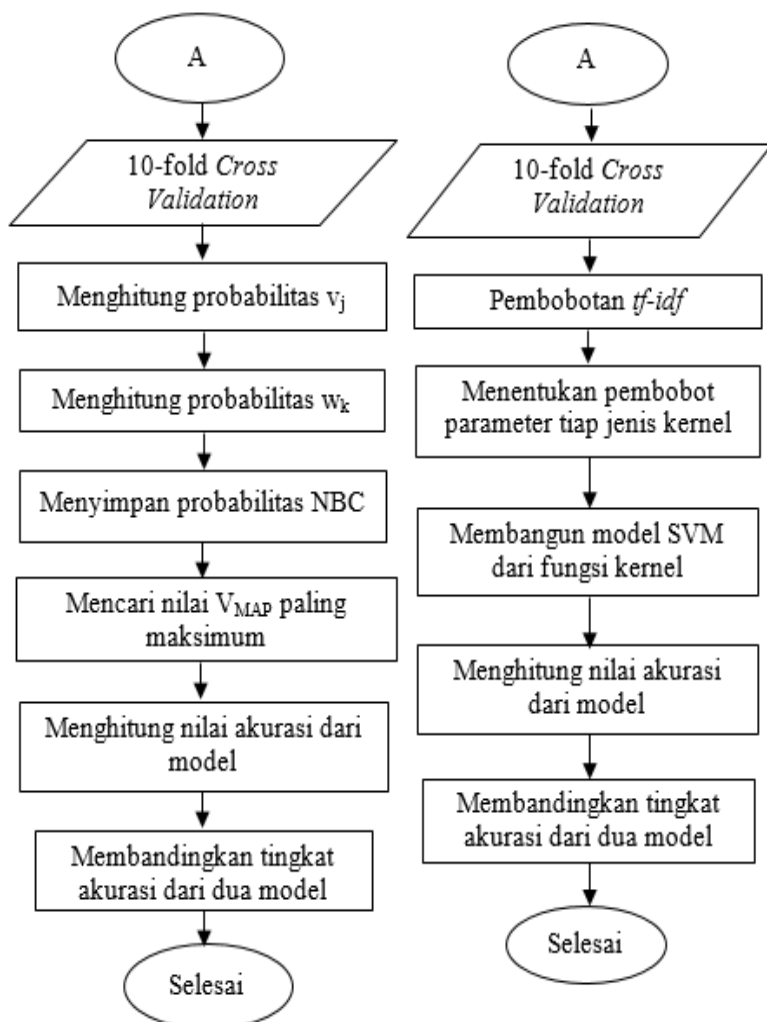
3. Klasifikasi teks menggunakan NBC dengan tahapan
 - a) Membentuk data dari hasil kata kunci dengan 10-fold *Cross Validation*.
 - b) Menghitung probabilitas dari V_j , dimana V_j merupakan klasifikasi sentimen, yaitu positif dan negatif.
 - c) Menghitung probabilitas kata w_k pada kategori v_j .
 - d) Model probabilitas NBC disimpan.
 - e) Menghitung probabilitas tertinggi dari semua kategori yang diujikan (V_{MAP}).
 - f) Mencari nilai V_{MAP} paling maksimum dan memasukkan *tweet* tersebut pada klasifikasi sentimen dengan V_{MAP} maksimum.
 - g) Menghitung nilai akurasi dari model yang terbentuk.
4. Klasifikasi teks menggunakan SVM dengan tahapan
 - a) Membentuk data dari hasil kata kunci dengan 10-fold *Cross Validation*.
 - b) Merubah teks menjadi vektor dan pembobotan kata dengan *tf-idf*.
 - c) Menentukan pembobot parameter pada SVM tiap jenis kernel.
 - d) Membangun model SVM menggunakan fungsi *Radial Basis Function* dan linier.
 - e) Menghitung nilai akurasi dari model yang terbentuk.
5. Membandingkan performansi metode NBC dan SVM berdasarkan tingkat akurasi ketepatan klasifikasi. Langkah analisis ini apabila digambar dengan diagram alir maka akan nampak seperti berikut:
6. Menentukan *Social Network Analysis* untuk menentukan pola jaringan - jaringan antar kata kunci disetiap calon Gubernur DKI Jakarta berdasarkan kata kunci yang muncul

pada data *tweet*. Didapatkan menggunakan *software* Gephi, yaitu *software* khusus dalam menganalisis *Social Network Analysis*

Diagram alir dari langkah analisis data pada penelitian ini disajikan dalam Gambar 3.1.



Gambar 3.1 Diagram Alir Praproses Teks



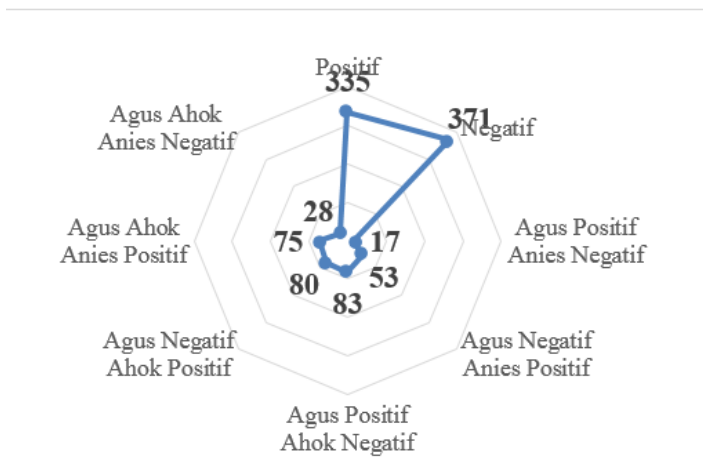
Gambar 3.2 Diagram Alir Membangun Model NBC dan SVM

BAB IV ANALISIS DAN PEMBAHASAN

Pada bab ini akan dibahas hasil analisis berdasarkan pengolahan data yang telah dilakukan. Metode yang digunakan dalam analisis adalah klasifikasi dengan *Naive Bayes Classifier* dan *Support Vector Machine* menggunakan data *Twitter API*. Sebelum menganalisis, dilakukan *pre-processing text*.

4.1 Karakteristik Data *Tweet* berdasarkan calon Gubernur DKI Jakarta

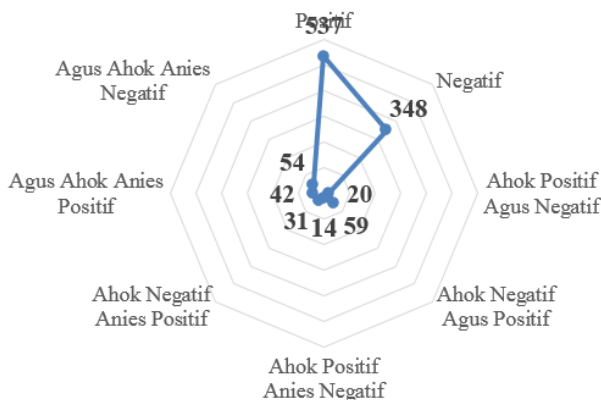
Jenis *tweet* yang di gunakan dalam pengambilan data menggunakan *Twitter API* adalah berdasarkan akun pribadi dari masing-masing calon Gubernur DKI Jakarta, yaitu @Agus-Yudhoyono, @basukibtp, dan @AniesBaswedan. Lalu dengan *golden training* atau berdasarkan dari persepsi peneliti maka diklasifikasikan ke dalam masing-masing 8 kelas untuk setiap calonnya. Karakteristik data *tweet* berdasarkan calon Gubernur DKI Jakarta disajikan pada Gambar 4.1.



Gambar 4.1 Karakteristik data *tweet* calon Gubernur DKI Jakarta Agus Yudhoyono

Gambar 4.1 menunjukkan bahwa dari 1154 *tweet* yang berhubungan dengan calon Gubernur DKI Jakarta Agus Yudhoyono cenderung dengan respon negatif dari pengguna *Twitter* sebesar 371 *tweet*, dan untuk hubungan antar kedua calon maupun ketiga calon mempunyai respon yang sedikit. Hal ini terjadi dikarenakan Agus Yudhoyono baru pertama terjun langsung di kancah politik setelah sebelumnya berada pada bidang militer, sehingga bermunculan *tweet-tweet* dengan respon negatif yang banyak dari *netizen*.

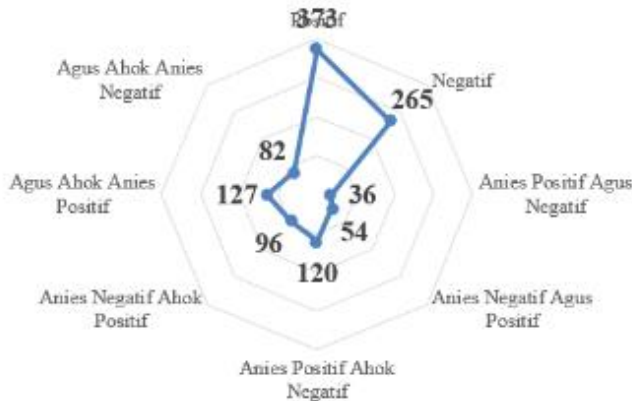
Karakteristik data *tweet* berdasarkan calon Gubernur DKI Jakarta Basuki Tjahaja Purnama (Ahok) dapat dilihat pada Gambar 4.2.



Gambar 4.2 Karakteristik data *tweet* calon Gubernur DKI Jakarta Agus Yudhoyono

Gambar 4.2 menunjukkan sebagian besar *tweet* yang mengacu pada calon Gubernur DKI Jakarta Ahok mendapatkan jumlah sentimen positif dan negatif yang memiliki jumlah yang jauh lebih besar dibandingkan dengan hubungan *tweet* dengan kedua calon Gubernur lainnya. Hal ini demikian karena beberapa pengguna *Twitter* masih memberikan elektabilitas yang tinggi untuk menjabat lagi dalam 1 periode kedepan. Untuk sentimen negatif, banyak *tweet* yang membawa isu SARA pada Ahok karena

diduga kuat dalam masalah kasus penistaan agama. Selain itu, karakteristik *netizen Twitter* terhadap calon Gubernur Anies Baswedan dapat dilihat pada gambar 4.3.



Gambar 4.3 Karakteristik data *tweet* calon Gubernur DKI Jakarta Anies Baswedan

Gambar 4.3 menunjukkan bahwa sedikitnya sentimen negatif dari pengguna *Twitter* yang mengarah kepada calon Gubernur Anies Baswedan. Dilihat dari *tweet* antar tiap calon Gubernur antar keduanya maupun ketiganya juga didapatkan hasil yang positif. Hal ini terkait respon *netizen* yang ingin mengetahui program terbaru dan slogan yang unik dari calon Gubernur tersebut.

4.2 Praproses Teks

Data *tweet* yang telah dikumpulkan kemudian dilakukan praproses teks yaitu *tokenizing*, *cleansing*, *case folding*, *stemming*, dan *stopwords*. Proses *tokenizing*, *case folding*, dan *stemming* menggunakan *software jupyter notebook* dengan bahasa pemrograman *python 2.7*. Praproses menggunakan salah satu *tweet* adalah sebagai berikut :

Tabel 4.1 Contoh Praproses Teks

1. <i>Tweet</i>	2. <i>Cleansing Tweet</i>
@jokoanwar: Ahok:Ini orang pikir ini luar negeri. Ini Kalijodo. Kami gak lagi jual program. Kami udah melakukan perubahan. https://t.co...	Ini orang pikir ini luar negeri. Ini Kalijodo. Kami gak lagi jual program. Kami udah melakukan perubahan
3. <i>Tokenizing, case folding, stemming</i>	4. <i>Stopwords</i>
ini orang pikir ini kalijodo kami tidak lagi jual program kami sudah melakukan perubahan	orang pikir kalijodo lagi jual program laku rubah

Pada Tabel 4.1 memperlihatkan sebuah artikel hingga didapatkan hasil akhir berupa teks yang telah melalui praproses dengan hasil akhir pada kolom *stopwords*. Untuk daftar *stop-words* pada contoh Tabel 4.2, digunakan daftar stopwords yang telah didefinisikan dalam tesis F. Tala yang berjudul “*A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia*” yang didapatkan di *packages* Sastrawi pada *python 2.7*. Sedangkan untuk daftar kata dasar didapatkan dari kamus besar bahasa Indonesia.

Tabel 4.2 Beberapa Contoh *Stopword* yang Digunakan

Stopword		
ada	...	pernah
adalah	...	yang

Berikut merupakan lima kata dengan frekuensi tertinggi untuk tiap sentimen pada *tweet* pada setiap calon Gubernur DKI Jakarta yang sudah ditentukan dari hasil *crawling data* menggunakan *Python*. Dari kata kunci yang didapat, secara keseluruhan sesuai dengan kondisi politik pemilihan Gubernur DKI Jakarta. Sehingga hasil dari menentukan frekuensi kata kunci ketiga Calon Gubenur DKI Jakarta berdasarkan pengguna *twitter* di dalam data *tweet* sudah sesuai.

Tabel 4.3 Frekuensi kata kunci calon Gubernur DKI Jakarta Agus

Positif		Negatif	
rtrw	85	rtrw	101
ketemu	81	program	46
jakarta	54	pimpin	35
fakta	40	jakarta	33
program	37	curiga	33

Tabel 4.3 menunjukkan frekuensi kata kunci yang mengarah kepada calon Gubernur Agus Yudhoyono, terlihat bahwa respon *netizen* pada sentimen positif terbesar yaitu rtrw, karena mengacu pada program unggulan dari calon Gubernur tersebut jika terpilih akan memberikan dana 1 miliar kepada rumah tangga di DKI Jakarta. Sementara rtrw juga berada pada sentimen negatif, karena hal tersebut juga masih banyak kekurangannya jika rumah tangga – rumah tangga di DKI Jakarta diberi 1 miliar. Kata pimpin dan curiga juga menghiasi respon negatif kepada calon Gubernur DKI Jakarta ini. Sehingga dikatakan bahwa masyarakat masih belum mendapatkan kepercayaan yang tinggi untuk menjabat sebagai Gubernur DKI Jakarta. Selanjutnya, kata kunci yang sering muncul pada *tweet* calon Gubernur DKI Jakarta Ahok.

Tabel 4.4 Frekuensi kata kunci calon Gubernur DKI Jakarta Ahok

Positif		Negatif	
kalijodo	196	perempuan	58
kerja	163	agama	45
program	154	fitnah	40
closing	69	jangan	37
gubernur	62	gubernur	37

Tabel 4.4 menunjukkan frekuensi kata kunci yang mengarah kepada calon Gubernur Basuki Tjahaja Purnama (Ahok), terlihat bahwa respon *netizen* pada sentimen positif terbesar yaitu kata kalijodo. Suatu tempat yang dulunya tempat prostitusi terbesar di utara Jakarta dirubah menjadi ruang terbuka hijau dan taman bermain anak disaat masa kepemimpinannya. Hal ini menunjukkan bahwa pengguna *twitter* masih ingin dipimpin satu periode lagi

oleh Ahok terlihat dari program kerjanya yang sudah terbukti serta tegas dalam mengatasi permasalahan DKI Jakarta. Sementara, respon negatif juga mendapatkan kata perempuan, dimana dengan tegasnya memimpin DKI Jakarta, ada beberapa *netizen* yang tidak suka dengan berperilaku kasar dengan perempuan dalam melayani warganya, lalu kata agama dan fitnah juga menjadi frekuensi kata terbesar, tentu berkaitan dengan kasus penistaan agama yang sangat di hubung-hubungkan dengan dunia politik. Sehingga dikatakan bahwa ada sebagian masyarakat yang masih ingin di pimpin, dan ada juga yang sudah menentang karena kasus yang didapatkan oleh calon Gubernur Ahok. Selanjutnya, kata kunci yang sering muncul pada *tweet* calon Gubernur DKI Jakarta Anies Baswedan.

Tabel 4.5 Frekuensi kata kunci calon Gubernur DKI Jakarta Anies

Positif		Negatif	
Jakarta	81	jakarta	60
Okoce	72	rumah	28
Baru	53	soal	27
Gubernur	46	mayoritas	25
Warga	45	nabung	17

Tabel 4.5 menunjukan frekuensi tertinggi dari calon Gubernur DKI Jakarta Anies Baswedan pada kata Jakarta, yaitu banyaknya respon positif untuk memimpin DKI Jakarta sebagai gubernur yang baru, sementara ada kata okoce, yaitu program unggulan dari calon Gubernur DKI Jakarta Anies yang sangat menarik perhatian warga ibukota, sehingga hasil putaran kedua Pilkada Gubernur DKI Jakarta berhasil diraih oleh Anies Baswedan. Sementara respon negatif dari pengguna *Twitter* yaitu berkaitan dengan program unggulannya yaitu rumah dengan DP 0%, kita ketahui kata-kata negatif yang muncul yaitu rumah, soal, mayoritas, serta nabung. Sehingga ketika masa tenang putaran 1, banyak *netizen* yang masih meragukan rumah dengan DP 0% tersebut bagi warga DKI Jakarta, selanjutnya diketahui frekuensi

kata kunci yang sering muncul dari hubungan tiap calon Gubernur antar keduanya maupun ketiga-tiganya dari sentimen yang berkembang di *Twitter*.

Tabel 4.6 Frekuensi kata kunci tiap calon Gubenur

	Agus – Ahok		Agus – Anies		Anies - Ahok	
Keras	72		baru	26	allah	59
Perempuan	60		jakarta	16	baru	33
Sby	58		gubernur	12	gubernur	30
Verbal	37		kerja	11	jangan	25
Laku	35		konsistensi	10	menang	21

Dari tabel 4.6 didapatkan terdapat hubungan dari kedua pasangan calon Gubernur DKI Jakarta. Terlihat ada hubungan antara kata-kata yang sering bermunculan seperti baru dan gubernur. Respon yang tinggi dari pengguna *Twitter* untuk memiliki gubernur yang baru dari ibukota ini. Selanjutnya ingin di ketahui hubungan antar ketiga calon Gubernur DKI Jakarta.

Tabel 4.7 Frekuensi kata kunci hubungan ketiga calon Gubernur DKI Jakarta

Positif		Negatif	
debat	80	debat	26
calon	25	bukan	20
gubernur	19	paslon	19
jakarta	18	gubernur	16
jadi	16	jakarta	16

Berdasarkan tabel 4.7, terlihat adanya hubungan yang tinggi antar ketiga calon Gubernur DKI Jakarta adanya kata debat dalam frekuensi terbesar membuktikan bahwa pengguna *Twitter* sangat tertarik dengan debat terakhir yang dibawakan oleh ketiga calon Gubernur DKI Jakarta.

Dalam uji *Naive Bayes Classifier* (NBC) maupun *Support Vector Machine* (SVM) data akan dibagi menjadi menggunakan *k*-

fold Cross-Validation, yaitu menggunakan 3, 5, 7, 10-*fold Cross Validation* yang diambil adalah ketepatan klasifikasi yang terbesar. Berikut merupakan hasil NBC dan SVM tiap calon Gubernur DKI Jakarta.

4.3 Klasifikasi Menggunakan *Naïve Bayes Classification*

Pada data pemecahan menjadi *k-fold cross validation* kelas pada sentimen telah diketahui sebelumnya. Dimana tujuan data *k-fold cross validation* adalah untuk menghasilkan model untuk mengetahui ketepatan klasifikasi. Seperti pada penjelasan sebelumnya data *tweet* akan menjadi lebih besar jika tidak dibatasi. Sehingga akan dicoba beberapa jumlah *k-fold cross validation* untuk menemukan model yang memberikan hasil yang paling optimum. Proses ini akan menggunakan seluruh data *tweet*. Berikut merupakan hasil uji data *tweet*.

Tabel 4.8 Ketepatan Klasifikasi Pembentukan Model NBC
Menggunakan Data *Tweet*

<i>k-fold CV</i>	Ketepatan Klasifikasi (%)
3	70.73
5	72.20
7	71.68
10	96.38

Tabel 4.9 memperlihatkan ketepatan klasifikasi dengan metode NBC pada data *tweet* menghasilkan nilai di atas 85% untuk 10-*fold cross validation*. Untuk angka bercetak tebal memperlihatkan bahwa dengan menggunakan 10-*fold cross validation* akan menghasilkan tingkat klasifikasi yang paling baik dibandingkan dengan jumlah *k-fold cross validation* lainnya. Gambar

4.4 akan membantu untuk menunjukkan ketepatan klasifikasi masing-masing.



Gambar 4.4 Persentase Akurasi Data Training NBC

Pada gambar 4.4 memperlihatkan dari 4 percobaan yang dilakukan, ketepatan klasifikasi *tweet* terus meningkat. Sehingga ketetapan klasifikasi terbaik berada pada *10-fold cross validation* dengan angka 96.38 %. Selanjutnya akan digunakan untuk mengukur performa NBC pada klasifikasi sentimen pemilihan Gubernur DKI Jakarta yang dibagi kedalam 7 kelas.

4.3.1 Pengukuran Performa NBC dengan Data *Tweet* 10-Fold Cross Validation

Pengukuran performa dilakukan dengan membandingkan antara hasil klasifikasi data *tweet* menggunakan model yang terbentuk pada *10-fold cross validation*. Langkah selanjutnya adalah pengukuran performa atau evaluasi model klasifikasi. Cara pengukuran tersebut menggunakan akurasi, *recall*, *precision* dan *F-Measure*. Akurasi ini didapatkan dari total dokumen / *tweet* yang teridentifikasi secara tepat, yang relevan, dan mengambil dari contoh milik kelas sasaran sentimen, positif maupun negatif. Berikut merupakan perhitungan untuk akurasi, *recall*, *precision*,

dan *F-Measure* untuk tiap klasifikasi *tweet* dalam ke-tiga calon Gubernur DKI Jakarta.

Tabel 4.9 Hasil Akurasi, *Precision*, *Recall*, dan *F-Measure* NBC pada Data *Tweet*

Klasifikasi Sentimen	Akurasi	Precision	Recall	F-Measure
Agus	56.20%	58.60%	56.20%	49.90%
Ahok	71.50%	74.60%	71.50%	68.80%
Anies	59.20%	57.50%	59.20%	56.40%
Agus – Ahok	74.80%	77.00%	74.80%	73.00%
Agus – Anies	63.10%	67.20%	63.10%	57.90%
Ahok – Anies	61.50%	60.90%	61.50%	57.40%
Agus – Ahok – Anies	67.90%	70.10%	67.90%	63.70%
Rata-rata	64.64%	66.80%	64.49%	61.01%

Berdasarkan rata-rata akurasi, *recall*, *precision*, dan *F-Measure* pada tabel 4.9 memperlihatkan hasil yang kurang baik. Hal ini dikarenakan data yang di pisah menjadi 7 klasifikasi sangat sedikit dan tentunya berpengaruh terhadap data *tweet* untuk membangun model. Selanjutnya, berikut untuk melihat *confusion matrix* yang didapat dari 7 klasifikasi sentimen.

Tabel 4.10 Hasil *confusion matrix* NBC pada Data *Tweet*

Klasifikasi Sentimen	Jumlah Tweet	True Positive	False Positive	True Negative	False Negative
Agus	706	65	39	332	270
Ahok	885	121	25	512	227
Anies	638	299	186	79	74
Agus – Ahok	242	48	9	133	52
Agus – Anies	160	83	53	6	18
Ahok – Anies	260	29	20	131	80
Agus – Ahok – Anies	408	228	115	16	49

Tabel 4.10 menunjukan bahwa adanya kesalahan dalam pembuatan *confusion matrix*, dikarenakan data yang digunakan *inbalance* atau tidak seimbang, bahkan untuk membangun satu

klasifikasi sentimen data yang dibentuk sangat sedikit. Salah satu yang bisa di atasi adalah menggabungkan ketujuh klasifikasi menjadi hanya tiga klasifikasi dengan masing-masing sentimen yang dimiliki oleh ketiga calon Gubernur DKI Jakarta. Hubungan antar kedua calon Gubernur atau ketiganya, akan dimasukkan kedalam kelas sentiment yang dimiliki oleh ketiga calon Gubernur. Sehingga, berikut merupakan perhitungan untuk akurasi, *recall*, *precision*, dan *F-Measure* untuk tiga klasifikasi sentimen *tweet*.

Tabel 4.11 Hasil Akurasi, *Precision*, *Recall*, dan *F-Measure* NBC pada Data *Tweet*

Klasifikasi Sentimen	Akurasi	Precision	Recall	F-Measure
Agus	83.30%	83.40%	83.30%	83.30%
Ahok	92.70%	92.80%	92.70%	92.70%
Anies	81.30%	81.50%	81.30%	81.00%
Rata-rata	85.77%	85.90%	85.77%	85.67%

Tabel 4.11 menunjukan tingkat akurasi paling tinggi dihasilkan oleh klasifikasi sentimen untuk calon Gubernur Ahok dengan 92.70%. Untuk ukuran gabungan dari *precision* dan *recall* yaitu *F-Measure* memperlihatkan bahwa klasifikasi sentimen *tweet* untuk ahok masih yang paling tinggi dengan 92.70%. Sementara untuk klasifikasi lainnya semua akurasi berada di atas 80%, klasifikasi ini lebih baik dibandingkan dengan klasifikasi sebelumnya yang rata-rata 60% karena data *tweet* yang *inbalance* dengan terbagi 7 klasifikasi sentiment. Selanjutnya diketahui tabel *confusion matrix* yang didapat dari 3 klasifikasi sentimen.

Tabel 4.12 Hasil *confusion matrix* NBC pada Data *Tweet*

Klasifikasi Sentimen	Jumlah <i>Tweet</i>	<i>True Positive</i>	<i>False Positive</i>	<i>True Negative</i>	<i>False Negative</i>
Agus	1347	545	91	577	134
Ahok	1416	528	29	785	74
Anies	1352	689	169	410	84

Tabel 4.12 menunjukan bahwa data sudah baik dalam pembuatan *confusion matrix*, dikarenakan data yang digunakan

sudah seimbang, terlihat untuk kesalahan klasifikasi terbesar terdapat pada sentimen calon Gubernur Anies, namun hasil tetap lebih baik dibandingkan hasil klasifikasi yang sebelumnya, selanjutnya adalah prediksi antar *tweet* calon Gubernur DKI Jakarta setelah mendapatkan *output* dengan metode Naïve Bayes Classification.

Tabel 4.13 Prediksi *tweet* antar Calon Gubernur DKI Jakarta

Prediksi Sentimen	<i>Tweet</i> Positif	<i>Tweet</i> Negatif
Agus	50 %	50 %
Ahok	57 %	43 %
Anies	57 %	43 %

Berdasarkan tabel 4.13 menunjukkan prediksi antar *tweet* calon Gubernur DKI Jakarta berdasarkan metode *Naïve Bayes Classification*. Untuk Calon Gubernur Agus Yudhoyono, mendapatkan prediksi *tweet* positif – negatif yang berimbang, hal ini menunjukkan bahwa pengguna *twitter* masih ragu dalam memilih calon Gubernur tersebut. Untuk pasangan Ahok dan Anies Baswedan, mendapatkan prediksi yang sama yaitu 57:43 untuk *tweet* positif–negatif. Sehingga, prediksi dengan menggunakan *Naïve Bayes Classification* tepat, sebab yang maju dalam putaran II pemilihan Gubernur DKI Jakarta adalah dari calon Gubernur tersebut.

4.4 Klasifikasi Menggunakan *Support Vector Machine*

Pembahasan pada SVM ini akan sama dengan pembahasan pada NBC, perbedaannya adalah ditambahkan parameter untuk SVM dengan kernel RBF yaitu C dan γ , kernel polynomial dengan parameter C , γ , dan p , sedangkan untuk kernel linier tidak ditambahkan parameter. Pembahasan pertama akan dilakukan dengan menggunakan kernel RBF dan kemudian dilanjutkan dengan menggunakan kernel polynomial dan juga kernel linier.

4.4.1 SVM Menggunakan Kernel RBF Pada Data *Tweet*

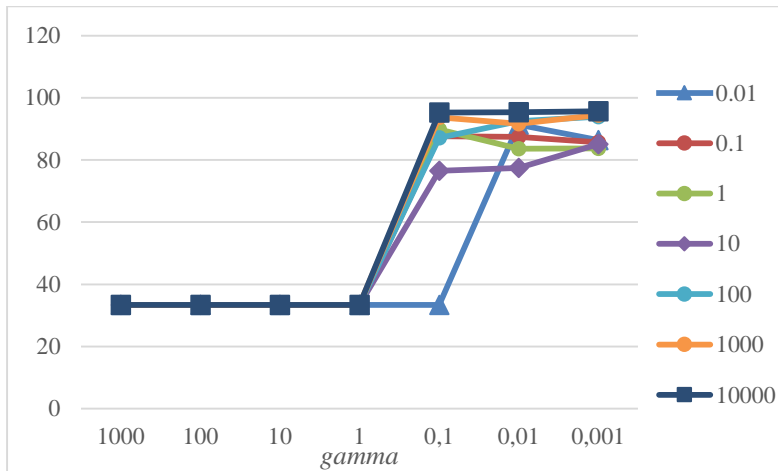
Pada mulanya data *tweet* dibagi menjadi 7 klasifikasi sentimen sama seperti pembahasan pada NBC. Tiap klasifikasi dicari ketepatan klasifikasi yang paling baik dengan atau tidak menggunakan parameter SVM untuk kernel *radial basis function* yaitu C dan $gamma$. Parameter C akan dicoba dari 10^{-2} hingga 10^4 (Huang, Lee, Lin, & Huang, 2007). Sedangkan untuk $gamma$ akan didapatkan melalui hasil percobaan agar mendapatkan hasil yang paling baik pada *10-fold cross validation*. Untuk mendapatkan model pada data ini akan digunakan data *tweet* yang terbanyak. Berikut merupakan hasil percobaan pada data *tweet* dengan menggunakan keseluruhan data *tweet* untuk melihat bagaimana pengaruh kedua parameter tersebut.

Tabel 4.14 Ketepatan Klasifikasi SVM Kernel RBF Menggunakan Data Training (%)

<i>Gamma</i>	1000	100	10	1	0.1	0.01	0.001
0.01	33.37	33.37	33.37	33.37	33.37	91.37	86.37
0.1	33.37	33.37	33.37	33.37	87.59	87.39	85.66
1	33.37	33.37	33.37	33.37	89.78	83.57	83.76
C 10	33.37	33.37	33.37	33.37	76.51	77.48	85.06
100	33.37	33.37	33.37	33.37	87.11	92.43	93.86
1000	33.37	33.37	33.37	33.37	93.75	91.68	94.45
10000	33.37	33.37	33.37	33.37	95.25	95.35	95.69

Berdasarkan Tabel 4.14 dapat dilihat bahwa dengan menggunakan parameter C dan $gamma$ pada percobaan SVM akan mempengaruhi hasil ketepatan klasifikasi *tweet* pada data *tweet*. Seperti yang terlihat pada Tabel 4.12 dengan $gamma$ 1000 hingga 1 didapatkan ketepatan klasifikasi pada data *tweet* sebesar 1% pada tiap parameter C . Nilai $gamma$ yang semakin mengecil mulai mempengaruhi ketepatan klasifikasi seperti yang terlihat pada $gamma$ 0,001 dimana untuk C 10^{-2} hingga 10^0 ketepatan klasifikasi naik menuju 100%. Semakin mengecil nilai $gamma$ semakin besar ketepatan klasifikasi sehingga perlu ditambahkan parameter C

yang lebih kecil. Pengaruh kedua parameter akan lebih jelas dengan gambar berikut ini.



Gambar 4.5 Ketepatan Klasifikasi SVM Kernel RBF Menggunakan Data *Tweet*

Pada Gambar 4.5 memperlihatkan bahwa dengan menggunakan parameter $C=10^4$ didapatkan hasil klasifikasi lebih baik dibandingkan apabila dengan menggunakan C yang lebih rendah. Parameter SVM dengan $C=10^4$ selalu stabil menghasilkan ketepatan klasifikasi tertinggi dari nilai γ 0,1, yang masih berada batas dari yang disarankan Huang dkk (2007). Pada data *tweet* dipilih menggunakan parameter $C=10^4$ dan $\gamma = 0,001$ atau angka yang bercetak tebal pada Tabel 4.12. Parameter tersebut dipilih berdasarkan hasil pada data *tweet* dan akurasi yang didapatkan cukup baik. Parameter lainnya seperti $\gamma = 1000$ hingga 0,01 tidak digunakan karena akurasi sangat rendah berkisar di angka 33,37%. Sebelum mengetahui performa klasifikasi, ditentukan angka yang tepat untuk *k-fold cross validation*.

Tabel 4.15 Ketepatan Klasifikasi Pembentukan Model NBC
Menggunakan Data *Tweet*

<i>k-fold CV</i>	Ketepatan Klasifikasi (%)
3	95,69
5	95,69
7	95,69
10	95,69

Tabel 4.15 memperlihatkan ketepatan klasifikasi dengan metode SVM pada data *tweet* menghasilkan nilai yang sama sebesar 95,69% untuk 10-fold cross validation. Untuk angka bercetak tebal memperlihatkan bahwa dengan menggunakan 10-fold cross validation akan menghasilkan tingkat klasifikasi yang paling baik dibandingkan dengan jumlah *k-fold cross validation* lainnya.

4.4.2 SVM Menggunakan Kernel Linier Pada Data *Tweet*

Kernel *polynomial* merupakan pengembangan dari kernel linier dengan menambahkan parameter γ dan p . Selama percobaan didapatkan hasil yang paling baik adalah dengan parameter $\gamma = 1$, dan $p = 1$. Isi dari parameter tersebut apabila dimasukkan akan sama dengan kernel linier. Maka selanjutnya akan dibahas mengenai ketepatan klasifikasi menggunakan kernel linier pada data *tweet*. Berikut merupakan hasil ketepatan klasifikasi oleh aplikasi untuk mendapatkan model dengan menggunakan parameter linier yaitu c dengan rentang nilai 10^{-2} hingga 10^4 . Hasil dari percobaan menunjukkan rentang nilai c tersebut menghasilkan hasil ketepatan klasifikasi yang hampir sama. Berdasarkan Tabel 4.16 memperlihatkan untuk SVM dengan menggunakan kernel linier untuk setiap *tweet* pada data *training* didapatkan nilai ketepatan sebesar 95,69%. Untuk selanjutnya akan digunakan kernel linier dengan parameter $c = 10^4$ untuk digunakan pada data *tweet* karena tiap *k-fold cross validation* memiliki ketepatan klasifikasi hampir 100% pada $c = 10^4$.

Tabel 4.16 Ketepatan Klasifikasi SVM Kernel Linier Menggunakan Data *Tweet*

		Ketepatan Klasifikasi (%)						
C		0.01	0.1	1	10	100	1000	10000
<i>k-fold CV</i>	3	89.53	93.70	94.49	93.88	93.57	94.47	95.69
	5	89.53	93.70	94.49	93.88	93.57	94.47	95.69
	7	89.53	93.70	94.49	93.88	93.57	94.47	95.69
	10	89.53	93.70	94.49	93.88	93.57	94.47	95.69

Setelah mengetahui bahwa ketepatan klasifikasi pada data *tweet* baik untuk SVM dengan kernel RBF maupun kernel linier, maka selanjutnya akan masuk tahap dengan menggunakan data *tweet* untuk tiap kernel.

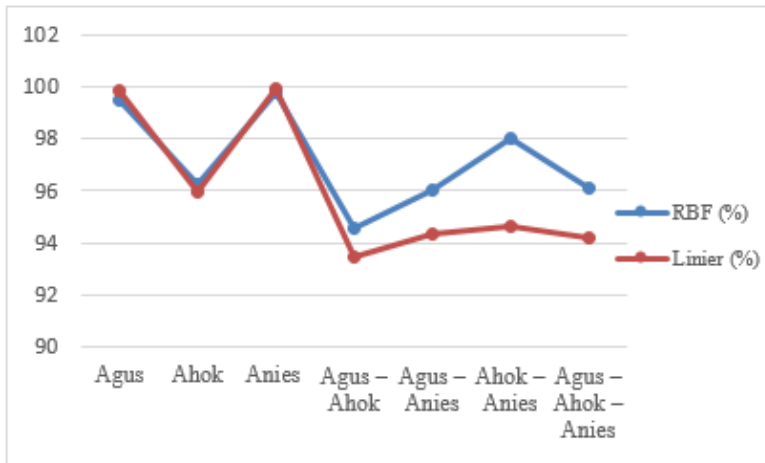
4.4.3 Penentuan Ketepatan Klasifikasi Sentimen dengan Kernel RBF dan Linier

Langkah yang digunakan sama dengan langkah yang digunakan pada NBC yaitu klasifikasi sentimen pada 7 kelas. Namun, belum diketahui dengan menggunakan model SVM yang terbentuk pada *tweet* baik untuk kernel RBF atau linier. Maka langkah selanjutnya dalam penelitian ini adalah menguji masing-masing model Berikut merupakan hasil klasifikasi sentimen dengan data *tweet* menggunakan model SVM yang telah terbentuk sebelumnya.

Tabel 4.17 Ketepatan Klasifikasi Sentimen SVM dengan RBF dan Linier

Klasifikasi Sentimen	RBF (%)	Linier (%)
Agus	99.49	99.81
Ahok	96.22	95.96
Anies	99.79	99.91
Agus – Ahok	94.53	93.47
Agus – Anies	96.06	94.33
Ahok – Anies	98.03	94.67
Agus – Ahok – Anies	96.13	94.20

Berdasarkan Tabel 4.17 akurasi yang terbentuk dalam menentukan klasifikasi sentimen SVM antara RBF dengan Linier sudah terlihat. Untuk lebih jelasnya akan ditampilkan pada Gambar 4.6



Gambar 4.6 Penentuan Ketepatan Klasifikasi SVM dengan Kernel RBF dan Linier

Gambar 4.6 dan Tabel 4.17 menunjukkan untuk kernel RBF bahwa data *tweet* yang diperoleh dari hasil 7 klasifikasi sentimen menghasilkan ketepatan klasifikasi yang lebih tinggi dibandingkan dengan kernel linier. Kemudian karena diketahui hasil 7 klasifikasi sentimen antara kernel RBF dan linier sama baiknya, maka akan dipilih menggunakan metode SVM dengan kernel RBF untuk dibandingkan dengan hasil pada NBC.

4.4.4 Pengukuran Performa SVM

Terdapat perbedaan hasil klasifikasi pada SVM dibandingkan dengan NBC. Terdapat beberapa klasifikasi sentimen yang tepat klasifikasinya pada saat NBC salah dan juga sebaliknya, selain itu terdapat prediksi dimana SVM dan NBC memasukkan kategori yang salah pada kategori yang berbeda diantara keduanya. Untuk SVM akan langsung dibahas mengenai

akurasi, *recall*, *precision*, dan *F-Measure* untuk tiap kategori berita. Berikut merupakan hasil dari model SVM dengan menggunakan data *tweet*.

Tabel 4.18 Hasil Akurasi, *Precision*, *Recall*, dan *F-Measure* SVM dengan Kernel RBF pada Data *Tweet*

Klasifikasi Sentimen	Akurasi	Precision	Recall	F-Measure
Agus	56.20%	58.60%	56.20%	49.90%
Ahok	72.30%	75.20%	72.30%	69.10%
Anies	59.60%	57.90%	59.60%	56.90%
Agus – Ahok	63.10%	67.20%	63.10%	57.90%
Agus – Anies	74.80%	77.00%	74.80%	73.00%
Ahok – Anies	61.50%	60.90%	61.50%	57.40%
Agus – Ahok – Anies	67.90%	70.10%	67.90%	63.70%
Rata-rata	65.06%	66.70%	65.06%	61.13%

Berdasarkan rata-rata akurasi, *recall*, *precision*, dan *F-Measure* pada tabel 4.18 memperlihatkan hasil yang tidak baik. Karena data yang di pisah menjadi 7 klasifikasi sedikit dan berpengaruh terhadap data *tweet* untuk membangun model. Selanjutnya, *confusion matrix* yang didapat dari 7 klasifikasi.

Tabel 4.19 Hasil *confusion matrix* NBC pada Data *Tweet*

Klasifikasi Sentimen	Jumlah Tweet	True Positive	False Positive	True Negative	False Negative
Agus	706	65	39	332	270
Ahok	885	129	26	511	219
Anies	638	299	184	81	74
Agus – Ahok	242	48	9	133	52
Agus – Anies	160	83	53	18	6
Ahok – Anies	260	29	20	131	80
Agus – Ahok – Anies	408	228	115	49	16

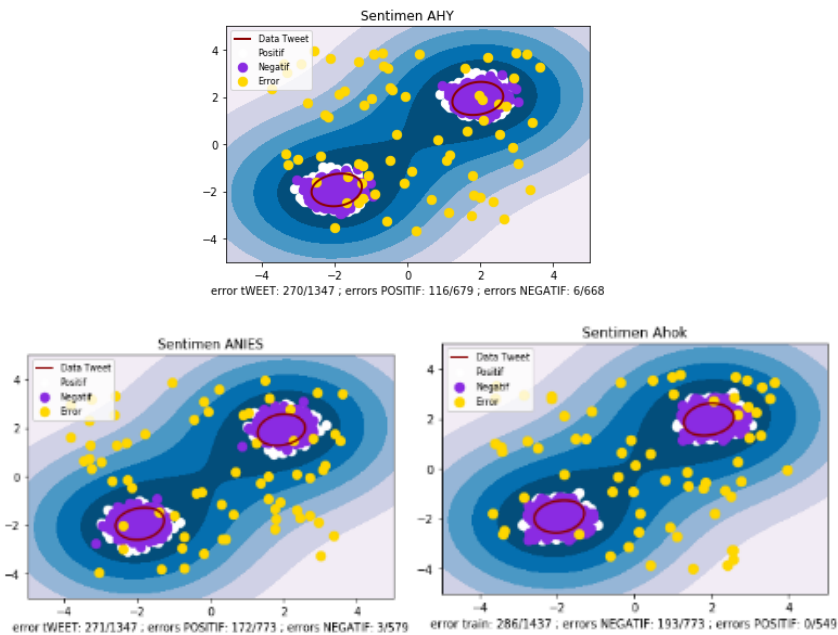
Tabel 4.19 menunjukan bahwa adanya kesalahan dalam pembuatan *confusion matrix*, dikarenakan data yang digunakan *inbalance* atau tidak seimbang, bahkan untuk membangun satu

klasifikasi sentimen data yang dibentuk sangat sedikit. Salah satu yang bisa di atasi adalah menggabungkan ketujuh klasifikasi menjadi hanya tiga klasifikasi dengan masing-masing sentimen yang dimiliki oleh ketiga calon Gubernur DKI Jakarta. Hubungan antar kedua calon Gubernur atau ketiganya, akan di masukan kedalam kelas sentiment yang dimiliki oleh ketiga calon Gubernur. Sehingga, berikut merupakan perhitungan untuk akurasi, *recall*, *precision*, dan *F-Measure* untuk tiga klasifikasi sentimen *tweet*.

Tabel 4.20 Hasil Akurasi, *Precision*, *Recall*, dan *F-Measure* SVM pada Data *Tweet*

Klasifikasi Sentimen	Akurasi	Precision	Recall	F-Measure
Agus	87.00%	96.71%	87.00%	91.60%
Ahok	88.55%	100%	88.55%	93.93%
Anies	87.85%	98.73%	87.85%	92.38%
Rata-rata	87.80%	98.48%	87.80%	92.64%

Hasil dari data *tweet* klasifikasi sentimen dengan SVM Kernel RBF menghasilkan nilai yang sangat baik. Klasifikasi Ahok menjadi yang terbesar dalam nilai Akurasi dan *F-Measure*. Sementara klasifikasi Anies menjadi yang terbesar dalam nilai *precesion*. Sedangkan untuk yang terendah klasifikasi Agus terjadi karena data *tweet* yang banyak terdapat dalam respon negative. Hal ini terjadi sebab banyak *netizen* akan ragu dengan elektabilitas dari calon Gubernur tersebut. Berdasarkan gambar 4.7 terlihat ilustrasi dari model SVM dengan Kernel RBF, dalam kasus ketiganya terlihat data sudah terpisah dengan *soft margin*. Sehingga didalam sentimen positif dan negatif, dapat memisahkan kata-kata tersebut dan memetakan data ke dimensi yang lebih tinggi, walaupun masih banyak kata-kata yang masih tersebar menjadi *error* data *tweet*.



Gambar 4.7 Ilustrasi SVM Kernel RBF ketiga calon Gubernur DKI Jakarta

Selanjutnya persamaan yang dibentuk dari model SVM kernel RBF. Pada kasus SVM ini terdapat 2 kelas sentimen dengan tiap *tweet* memiliki jumlah *support vector* yang berbeda pada gambar 4.21.

Tabel 4.21 Persamaan Model SVM Kernel RBF

Klasifikasi Sentimen	Persamaan
Agus	$\sum_{i=1}^{1347} \exp(-10000 -0.0472 + (-0.0385) + \dots + (0.0272) ^2)$
Ahok	$\sum_{i=1}^{1416} \exp(-10000 -0.0265 + (-0.0375) + \dots + (0.0460) ^2)$
Anies	$\sum_{i=1}^{1352} \exp(-10000 -0.0271 + (-0.0384) + \dots + (-0.0609) ^2)$

4.5 Perbandingan Hasil Klasifikasi Antara NBC dan SVM

Setelah mengetahui hasil masing-masing ketepatan klasifikasi pada kedua metode maka langkah selanjutnya adalah membandingkan. Berikut merupakan perbandingan antara kedua metode berdasarkan akurasi, *precision*, *recall*, dan *F-Measure*.

Tabel 4.22 Perbandingan Metode NBC dan SVM

Metode	Akurasi	Precision	Recall	F-Measure
NBC	85.77%	85.90%	85.77%	85.67%
SVM	87.80%	98.48%	87.80%	92.64%

Melihat hasil dari Tabel 4.22 maka untuk semua cara pengukuran performa baik akurasi, *precision*, *recall*, dan *F-Measure* SVM kernel linier dan RBF lebih baik dari NBC. Selain itu secara waktu saat menggunakan aplikasi *python* jauh lebih cepat untuk mendapatkan hasil daripada NBC.

4.6 Hubungan Antar Kata Kunci Ketiga Calon Gubernur DKI Jakarta

Setelah mengetahui model terbaik yang sudah dibentuk, selanjutnya adalah mencari tahu hubungan antar kata kunci setiap calon Gubernur DKI Jakarta yang sudah dibentuk. Hal ini bertujuan untuk menekankan pada interaksi antar entitas didalam calon Gubenur tersebut, dengan kata lain, akan lebih banyak membahas hubungan antar calon Gubernur daripada dengan kata kunci yang telah dimiliki keduanya atau ketiganya sekaligus. Pola interaksi antar entitas akan memberikan informasi baru. Khususnya mengetahui kelebihan dan kelemahan calon lain.

4.6.1 *Wordcloud* Interaksi antar Ketiga Calon Gubernur DKI Jakarta

Salah satu analisis yang tepat untuk menampilkan kata-kata populer yang muncul dari setiap calon Gubernur adalah dengan *wordcloud*. Semakin sering kata yang muncul dalam teks yang dianalisis, semakin besar ukuran kata yang muncul dalam

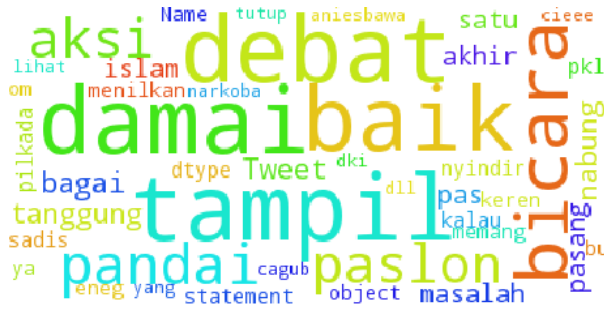
Gambar 4.9 *Wordcloud* Anies - Agus

Pada Gambar 4.9 menampilkan *wordcloud* kata kunci untuk calon Gubernur Anies - Agus. Dari 50 kata yang ditampilkan ada kata kunci yang memperlihatkan ukuran yang besar seperti fpi, Jakarta, transportasi, debat. Dari ukuran *wordcloud* diatas juga memberikan makna frekuensi kata yang sering muncul antara interaksi Anies - Agus. Hasil yang di-dapatkan sangatlah sesuai dengan hubungan yang terjadi pada kenyatannya. Keduanya merupakan calon Gubernur baru dengan latar belakang yang berbeda sehingga ingin mengetahui cara-cara mengatasi permasalahan di DKI Jakarta. Sehingga hasil dari interaksi *wordcloud* tersebut sudah sesuai. Selanjutnya hubungan antara calon Gubernur DKI Jakarta Ahok – Agus.



Gambar 4.10 *Wordcloud* Ahok - Agus

Pada Gambar 4.10 menampilkan *wordcloud* kata kunci untuk calon Gubernur Ahok - Agus. Dari 50 kata yang ditampilkan ada kata kunci yang memperlihatkan ukuran yang besar seperti perempuan, Jakarta, cagub, Jokowi, sby. Dari ukuran *wordcloud* diatas juga memberikan makna frekuensi kata yang sering muncul antara interaksi Ahok - Agus. Hasil yang di-dapat-kan sangatlah sesuai dengan hubungan yang terjadi pada kenyatannya. Keduanya merupakan calon Gubernur baru dengan latar belakang kedekatan dengan orang penting di Indoneisa yaitu Jokowi dan sby. Sehingga hasil dari interaksi *wordcloud* tersebut sudah sesuai. Selanjutnya hubungan antara calon Gubernur DKI Jakarta Ahok – Agus. Selanjutnya, hubungan antar ketiga calon



Gambar 4.11 Wordcloud Ahok – Agus – Anies

Pada Gambar 4.11 menampilkan *wordcloud* kata kunci untuk calon Gubernur Ahok – Agus – Anies. Dari 50 kata yang ditampilkan ada kata kunci yang memperlihatkan ukuran yang besar seperti aksi, damai, debat, tampil, baik. Dari ukuran *word-cloud* diatas juga memberikan makna frekuensi kata yang sering muncul antara interaksi Ahok – Agus – Anies. Hasil yang didapatkan sangatlah sesuai dengan hubungan yang terjadi pada kenyatannya. Ketiganya tampil apik di debat terakhir calon Gubernur DKI Jakarta, sehingga *netizen* memberikan apresiasi yang lebih kepada ketiga calon Gubernur tersebut.

4.7 Social Network Analysis (SNA)

Setelah mengetahui *wordcloud* antar hubungan kedua maupun ketiga calon Gubernur DKI Jakarta tersebut, selanjutnya adalah menentukan *sosial network analysis* untuk mengetahui pemetaan dan pengukuran hubungan – hubungan serta pola antara ketiga calon Gubernur tersebut dengan pengolahan jaringan struktur entitas. Simpul jaringan yang membentuk ketiga calon Gubernur tersebut merupakan dasar dari kata-kata / entitas yang mengarah kepada calon Gubernur tersebut. Dari ketiga Calon Gubernur tersebut, ada kata-kata yang identik menggambarkan Calon Gubernur tersebut, seperti Anies menghubungkan kata prabowo, strategi, serta okoc yang sesuai dengan kondisi politik yang sebenarnya. Sementara dari kata Ahok dihu-bungkan kata-

muncul di gambar tersebut meng-gunakan tipe *Fruchterman-Reingold*. Tipe tersebut merupakan tipe yang umum digunakan dalam pembuatan *social network analysis*. Berikut merupakan deskripsi dari *SNA* tipe *Fruchterman-Reingold*.

Tabel 4.23 Deskripsi *SNA* tipe *Fruchterman-Reingold*

<i>Degree Centrality</i>	<i>Closeness Centrality</i>	<i>Betweenness Centrality</i>	<i>Page rank</i>	<i>Eigenvector Centrality</i>
6	0.865	0	0.019	0.073

Dari Tabel 4.23 di atas menunjukan rata-rata dari keseluruhan node-node yang terbentuk, terdapat 53 node yang membentuk *Social Network Analysis*, berdasarkan *degree centrality* membentuk 6 yang menjadi node yaitu sentimen tiap calon gubernur. Sementara angka 0.855 pada *closeness centrality* menunjukkan jarak antara rata-rata node dengan semua node yang lain di jaringan. Hasil tersebut merupakan tinggi yang berarti setiap node mempunyai pengaruh antar node lainnya yang terhubung sebesar 0.865. *Betweenness Centrality* menunjukkan arti node sebagai persimpangan antar node lainnya, karena bernilai 0, maka tidak ada persimpangan node yang ada karena nilai *closeness centrality* yang tinggi. Nilai *eigenvector centrality* menunjukkan nilai 0.073, yang artinya bobot yang lebih tinggi dari pada bobot node yang memiliki keterhubungan yang tinggi sebesar 0.073. Terakhir adalah *Page Rank*, merupakan suatu distribusi yang digunakan *google* dalam menentukan suatu *page*. *SNA* dapat digunakan pula untuk jaringan yang berbentuk *graph* berarah, mendapatkan nilai yang rendah sebesar 0.019, karena semua berasal dari satu sumber yaitu situs *twitter*, maka nilai yang didapatkan rendah.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah sebelumnya didapatkan hasil dan pembahasan untuk klasifikasi *tweet* analisis sentimen pemilihan Gubernur DKI Jakarta menggunakan metode NBC dan SVM. Berikut merupakan kesimpulan yang didapatkan.

1. Dari *radar chart* yang sudah terbentuk kedalam 7 kelas sentimen, didapatkan hasil yang lebih banyak pada sentimen tentang calon itu sendiri, dikarenakan data yang berhubungan dengan kedua pasangan calon dan ketiganya hanya sedikit.
2. Metode *Naive Bayes Classifier* dapat melakukan klasifikasi *teks* dengan cukup baik karena hanya dengan 3 klasifikasi sentimen, setelah sebelumnya data tidak baik dengan 7 klasifikasi sentimen karena data *inbalance*. Hasil yang didapatkan pada saat data *tweet* dengan *10-fold cross validation* pada masing-masing pengukuran performa akurasi, *precision*, *recall*, dan *F-Measure* sebesar 85.77%; 85.90%; 85.77%; 85.67%.
3. Metode *Support Vector Machine* antara kernel RBF dan kernel linier didapatkan hasil kernel linier sama baiknya dengan kernel RBF, sehingga akurasi tertinggi yang digunakan adalah kernel RBF. Menggunakan data *tweet* dengan *10-fold cross validation*, didapatkan untuk tiap pengukuran performa akurasi, *precision*, *recall*, dan *F-Measure* adalah 87.80%; 98.48%; 87.80%; 92.64%.
4. Perbandingan antara kedua metode NBC dan SVM didapatkan hasil SVM kernel RBF lebih baik dibandingkan dengan NBC.
5. Hasil yang didapatkan dari hubungan antar kedua maupun ketiga calon Gubernur DKI Jakarta menghasilkan kata kunci yang sesuai dengan kondisi nyata. Dalam *wordcloud*,

seperti nista, agama, hingga nama tokoh seperti jokowi dan sby. Untuk hasil *SNA* didapatkan hasil yang tinggi untuk *degree centrality* sebesar 0.865 yang menunjukkan pengaruh antar node kata kunci dengan kata kunci yang lain.

5.2 Saran

Saran untuk penelitian yang akan datang adalah.

1. Pada penelitian klasifikasi *tweet* ini sangat membutuhkan klasifikasi sentiment yang cukup besar agar data yang di gunakan sama baik.
2. Pada proses *steeming*, perlu menambahkan kata kunci khusus untuk Bahasa *slang*, atau bahasa sehari-hari, karena kata tiap kata didalam *tweet* pasti menggunakan bahasa sehari-hari.

DAFTAR PUSTAKA

- Arifiyanti, A. A. (2014). *Klasifikasi Artikel Berita Berbahasa Indonesia Berbasis Naive Bayes Classifier Menggunakan Confix-Stripping Stemmer*. Surabaya: Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.
- Ariadi, D. & Fithriasari, K. (2015). Klasifikasi Berita Indonesia Menggunakan Metode Naïve Bayesian Classification dan Support Vector Machine dengan Confix Stripping Stemmer. *Jurnal Sains dan Seni ITS*, 4(2), 2337-3520.
- Asian, J. A. (2007). Stemming Indonesian : A Confix-Stripping Approach. *ACM Trnsactions on Asian Language Information Processing (TALIP)*, 6(4), 1-33. <http://doi.acm.org/10.1145/1316457.1316459>.
- Dragut, E., Fang, F., Sistla, P., Yu, C., & Meng, W. (2009). Stop Word and Related Problems in Web Interface Integration. *VLDB Endowment*. <http://doi.acm.org/10.14778/1687627.1687667>
- Durajati, C., & Gumelar, A. B. (2012). Pemanfaatan Teknik Supervised Untuk Klasifikasi Teks Bahasa Indonesia. *Jurnal Link Vol 16/No. 1*, 1-8. ISSN 1858 - 4667
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning* (Vol. 1). Springer, Berlin: Springer series in statistics.
- Guduru, N. (2006). *Text Mining With Support Vector Machines And Non-Negative Matrix Factorization Algorithms*. University Of Rhode Island. [doi:10.1109/ICCV.2007.4409066](http://doi.org/10.1109/ICCV.2007.4409066).
- Hamzah, A. (2012). Klasifikasi Teks dengan Naïve Bayes Classifier (NBC) untuk Pengelompokan Teks Berita dan Abstract Akademis. In *Prosiding Seminar Nasional Apikasi Sains & Teknologi (SNAST) Periode III*, p. B269B277. Yogyakarta.

- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques: Concepts And Techniques*. Elsevier.
- Hassan, S., Rafi, M., Shaikh, M.S. (2012). Comparing SVM and Naive Bayes classifiers for text categorization with Wikitology as knowledge enrichment. New York : 2011 IEEE 14th International. doi:10.1109/INMIC.2011.6151495.
- Hearst, M. A. (1997, July). Text Data Mining: Issues, Techniques, and The Relationship to Information Access. In *Presentation notes for UW/MS workshop on data mining* (pp. 112-117).
- Hidayat, W. (2014). *Pengguna Internet Indonesia Nomor Enam Dunia* . Retrieved Februari 2, 2015, from [kompas.com: http://tekno.kompas.com/read/2014/11/24/07430087/Pengguna.Internet.Indonesia.Nomor.Enam.Dunia](http://tekno.kompas.com/read/2014/11/24/07430087/Pengguna.Internet.Indonesia.Nomor.Enam.Dunia)
- Hotho, A., Nurnberger, A., & Paass, G. (2005). *A Brief Survey of Text Mining*. Kassel: University of Kassel. doi: 10.15680/IJIRCCE.2016. 0404040. 6564.
- Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2010). *A Practical Guide to Support Vector Classification*. Taiwan: Department of Computer Science National Taiwan University. doi:10.1.1.224.4115.
- Huang, C.-M., Lee, Y.-J., Lin, D. K., & Huang, S.-Y. (2007). Model Selection For Support Vector Machines Via Uniform Design. *Computational Statistics & Data Analysis*, 335-346. doi:10.1016/j.csda.2007.02.013.
- Kurniawan, B., Effendi, S., & Sitompul, O. S. (2012). Klasifikasi Konten Berita Dengan Metode. *Jurnal Dunia Teknologi Informasi Vol. 1, No. 1*, 14-19.
- Komisi Pemilihan Umum DKI Jakarta. 2017. *KPU Provinsi DKI Selesaikan Rekapitulasi Perhitungan Suara Hari Ini*. http://kpujakarta.go.id/view_berita/kpu_provinsi_dki_sel_esaikan_rekapitulasi_penghitungan_suara_hari_ini diakses pada tanggal 8 Maret 2017

- Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X., & Wang, S. (2011). An Improved Particle Swarm Optimization for Feature Selection. *Journal of Bionic Engineering*, 8(2), 191–200. doi:10.1016/s1672-6529(11)60020-6.
- Miller, T. (2005). *Data and Text Mining A Business Application*. New Jersey, USA: Prentice Hall.
- Monarizqa, N., Nugroho, L, E., Hantono, S, B. (2014). Penerapan Analisis Sentimen Pada Twitter Berbahasa Indonesia Sebagai Pemberi Rating. Yogyakarta : Jurnal Penelitian Teknik Elektro dan Teknologi Informasi. doi:10.328.21.22.
- Pak, A., Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In : ELRA(European Language Resource Association). Malta : International Language Resources a Evaluation (LREC'10). doi: 10.17148/IJARCCE.2016.51274.
- Rish, I. (2006). An empirical study of The Naive Bayes Classifier. *International Joint Conference on Artificial Intelligence*. doi:10.1007/978-3-319-50127-7_27.
- Saraswati, N. W. (2011). *Text Mining Dengan Metode Naive Bayes Classifier dan Support Vector Machine untuk Sentiment Analysis*. Denpasar: Program Pascasarjana Universitas Udayana.
- Sunni, I., Widyanoro, H, D. (2012). Analisis Sentimen dan Ekstraksi Topik Penentu Sentimen pada Opini Terhadap Tokoh Publik. Bandung : Jurnal Sarjana Institut Teknologi Bandung Bidang Teknik Elektro dan Informatika.
- Tala, F. Z. (2003). *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Netherlands: Master of Logic Project. Institute for Logic, Language and Computation, Universiteit van Amsterdam. doi:10.1145/1316457.1316459.
- Tan, P. N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining*. Boston: Pearson Education.

- Weiss, S. M. (2010). *Text mining: Predictive Methods for Analyzing Unstructured Information*. New York: Springer.
- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining Practical Machine Learning Tools and Techniques*. USA: Elsevier.
- Wicaksono, A. I., Nio, E., & Myaeng, S. H. Unsu (2013) Approach for Sentiment Analysis on Indonesian Movie. <http://dx.doi.org/10.15242/IIEC>.

DAFTAR LAMPIRAN

LAMPIRAN 1 Data *Tweet* Asli dari *Twitter* API (Contoh Data *Tweet* Ahok)

Tweet	Klasifikasi
Ahok menampilkan taman Kalijodoh yang sdh bertaraf internasional, paslon 1 dan 3 jangan gunakan data2 bohong utk sekedar...	7
Closing debat pilkada DKI pak ahok. MEREKA ITU ORANG TUA YG DIDIK ANAK AGAR JADI BAIK. tiba2 Om dan tante Paslon 1 & 3 Datang godain anak2..	7
ahok gagal bikin klimaks di debat terakhir ;(2
Fix lah ahok-djarot juara	1
Pemenang debat menurut ane nih yaDebat #1: AhokDebat #2: AhokDebat #3: AniesNgga tau juga sih ehehe.	1
Closing statement ahok sedikit emosional. Zonk !!	2
Ahok: Om tante jangan gampanggu orang tua yang udah mendidik anaknya IHY! \ ½\, #DebatFinalPilkadaJKT	2
#Aksi211 Nusron Akhirnya Tanggapi Hujatan Ahok kepada KH Maruf Amin, Komentarnya Sungguh Tak..... https://t.co/QSBTQtknwn	2
Closing statemen Basuki Jarot tentang Kalijodo menggetarkan. Ini kerja nyata.. Ahok ciptakan goll di masa injury time..	1
.	
.	
.	
@jokoanwar: Ahok:Ini orang pikir ini luar negeri. Ini Kalijodo. Kami gak lagi jual program. Kami udah melakukan perubahan. https://t.co...	1

Keterangan Klasifikasi Teks:

- | | |
|------------------------------|--------------------------------|
| 1. Ahok Positif | 5. Ahok Positif Anies Negatif |
| 2. Ahok Negatif | 6. Ahok Negatif Anies Positif |
| 3. Ahok Positif Agus Negatif | 7. Ahok – Agus – Anies Positif |
| 4. Ahok Negatif Agus Positif | 8. Ahok – Agus – Anies Negatif |

LAMPIRAN 2 Data *Tweet* Setelah Praproses Teks (Contoh Data *Tweet* Ahok)

Klasifikasi	TWEET
	tampil taman kalijodoh yang sdh taraf internasional paslon dan jangan guna data2 bohong utk dar closing debat pilkada dki pak mereka itu orang tua yg didik anak agar jadi baik tiba2 om dan tante paslon datang godain anak2
NEGATIF	gagal bikin klimaks di debat akhir
POSITIF	fix lah juara
POSITIF	menang debat turut ane nih yadebat debat debat aniesngga tau juga sih ehehe
NEGATIF	closing statement sedikit emosional zonk
NEGATIF	om tante jangan ganggu orang tua yang udah didik anak ihy nusron akhir tanggap hujat kepada kh maruf amin komentar sungguh tak
NEGATIF	closing statemen tentang kalijodo getar ini kerja nyata cipta goll di masa injury time
	.
	.
	.
POSITIF	ini orang pikir ini luar negeri ini kalijodo kami gak lagi jual program kami udah laku ubah

LAMPIRAN 3 *Syntax Pre-procesing Text Python (lanjutan)*

```

import csv
import sys
import string
import nltk
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.tokenize import TweetTokenizer
from nltk.corpus import stopwords
import json,re

#memanggil data
data = pd.read_csv("D:/AHOK1DAN2PRE.csv")
data.head(20)

#cleansing tweet
with open('D:/AHOK1DAN2PRE.csv') as namescsv:
    namereader = csv.reader(namescsv)
    for row in namereader:
        for cell in row:
            #menghapus username Twitter
            cell = re.sub( r'^([^\w])@(\w{1,15})\b', "", cell)
            #menghapus link internet
            cell = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', "",cell)
            #menghapus hashtag
            cell = re.sub(r"#(\w+)", "",cell)
            #menghapus karakter spesial
            cell = re.sub(' +', '',cell)
            cell = re.sub('[^A-z0-9 -]', "", cell)
            #menghapus nama calon gubernur
            cell = re.sub('(((Ahok-Djarot)|(ahok-djarot)|(ahok-
jarot)|(Ahok)|(ahok)|(AHOK)|(Djarot)|(djarot)|(Pak)))', "",cell)
            print cell

```

LAMPIRAN 4 Frekuensi Kata kunci yang muncul

X ₁	X ₂	X ₃	X ₄
Rtrw (105)	Kalijodo (196)	Jakarta (81)	Keras
Ketemu (81)	Kerja (163)	Keras (72)	Perempuan (60)
Jakarta (54)	Program (154)	Baru (53)	Sby (58)
Fakta (40)	Closing (69)	Gubernur (46)	Verbal (37)
Program (37)	Gubernur (62)	Warga (45)	Laku (35)
Pimpin (35)	Perempuan (58)	Korban (39)	Kaum (31)
Curiga (33)	Agama (45)	Lindung (34)	Fitnah (15)
.	.	.	.
.	.	.	.
.	.	.	.
Timeses (1)	Orasi (1)	Bangsa (1)	Kreasi (1)

X ₅	X ₆	X ₇
Baru (26)	Allah (59)	Debat (80)
Jakarta (16)	Baru (33)	Paslon (25)
Gubernur (12)	Gubernur (30)	Gubernur (19)
Soal (17)	Jangan (25)	Jakarta (18)
Pasang (16)	Dki (24)	Orang (16)
Kerja (11)	Menang (21)	Closing (15)
Konsentrasi (10)	Butuh (21)	Dki (13)
.	.	.
.	.	.
.	.	.
Posisi (1)	Sederhana (1)	prabowo (1)

Keterangan :

X₁ = Agus YudhoyonoX₂ = AhokX₃ = Anies BaswedanX₄ = Agus – AhokX₅ = Agus – AniesX₆ = Anies - AhokX₇ = Ahok – Agus – Anies

LAMPIRAN 5 *Syntax Python mencari frekuensi dokumen*

```
import re
import string
frequency = { }
document_text = open('D:/AHOKNEGATIF.csv', 'r')
text_string = document_text.read().lower()
match_pattern = re.findall(r'\b[a-z]{3,15}\b', text_string)

for word in match_pattern:
    count = frequency.get(word,0)
    frequency[word] = count + 1

frequency_list = frequency.keys()

for words in frequency_list:
    print words, frequency[words]
```

LAMPIRAN 6 *Syntax Python SVM Kernel RBF*

```
import collections
import nltk.classify.util, nltk.metrics
from nltk.classify import NaiveBayesClassifier, MaxentClassifier,
SklearnClassifier

import csv
from sklearn import cross_validation
from sklearn.svm import LinearSVC, SVC
from sklearn import svm

import random
from nltk.corpus import stopwords
import itertools
from nltk.collocations import BigramCollocationFinder
from nltk.metrics import BigramAssocMeasures
```

(lanjutan)

#definisi prediksi kata

def predict(training_data, test_data, test_label, classifier):

method for predicting new data

count_vectorizer = CountVectorizer(binary=True)

count_vectorizer.fit_transform(training_data)

test_data = count_vectorizer.transform(test_data)

test_data_clean =

TfidfTransformer(use_idf=True).fit_transform(test_data)

prediction = classifier.predict(test_data_clean)

classification_report(test_label, prediction)

acc = accuracy_score(test_label, prediction)

return acc

print "The accuracy score of new data is

{:.2% }".format(accuracy_score(test_label, prediction))

#mendefinisikan cross validation

def learn_model(training_data, training_label, classifier):

count_vectorizer = CountVectorizer(binary=True)

train = count_vectorizer.fit_transform(training_data)

tfidf_train = TfidfTransformer(use_idf=True).fit_transform(train)

data_train, data_test, target_train, target_test =

 cross_validation.train_test_split(tfidf_train, training_label,
 test_size=0.9,

random_state=42)

classify = classifier.fit(data_train, target_train)

 scores = cross_validation.cross_val_score(classify, data_test,
 target_test, cv=10)

return scores.mean()

 # print("Accuracy with 10-fold validation: %0.2f (+/- %0.2f)" %
(scores.mean(), scores.std() * 2))

(lanjutan)

```
#membaca data
df = pd.read_csv('D:/ANIESFIX.csv', encoding = "ISO-8859-1")
text_train = df['TWEET'] #ambil kolom text
df_train = text_train
df_label = df['Klasifikasi Teks'] #ambil kolom label
print df.head()

#model SVM Kernel RBF
svm_classifier = SVC(kernel='rbf', C=10000, gamma=0.001)

#menentukan performa klasifikasi akurasi , presisi, dan F-measure
from sklearn.metrics import accuracy_score
df = pd.read_csv('D:/ANISFIX.csv', encoding = "ISO-8859-1")
text_train = df['yt'] #ambil kolom text
y_pred = text_train
y_true = df['yp'] #ambil kolom label
accuracy_score(y_pred, y_true)

from sklearn.metrics import precision_score
precision_score(y_true, y_pred, average='weighted')

from sklearn.metrics import f1_score
f1_score(y_true, y_pred, average='weighted')
```

LAMPIRAN 7 *Syntax Wordcloud di Python*

```

import numpy as np # linear algebra
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline

from subprocess import check_output
from wordcloud import WordCloud, STOPWORDS

#mpl.rcParams['figure.figsize']=(8.0,6.0)  #(6.0,4.0)
mpl.rcParams['font.size']=16              #10
mpl.rcParams['savefig.dpi']=100           #72
mpl.rcParams['figure.subplot.bottom']=.1

stopwords = set(STOPWORDS)
data = pd.read_csv("D:\A-A-A.csv")

wordcloud = WordCloud(
    background_color='white',
    stopwords=stopwords,
    max_words=100,
    max_font_size=60,
    random_state=42
).generate(str(data['Tweet']))

print(wordcloud)
fig = plt.figure(1)
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
fig.savefig("word1.png", dpi=900)

```

LAMPIRAN 8 Hasil *Social Network Analysis* menggunakan *Gephi 0.9.1*

<i>Node</i>	<i>Edge</i>	<i>Degree</i>	<i>Clossnes</i>	<i>Betweenes</i>	<i>Page Rank</i>	<i>Eigen vector</i>
JOKOWI	JOKOWI	5	1	0	0.01055	0
banjir	banjir	5	1	0	0.01055	0
macet	macet	4	1	0	0.01055	0
kalijodo	kalijodo	3	1	0	0.01055	0
nista	nista	3	1	0	0.01055	0
jangan	jangan	2	1	0	0.01055	0
strategi	strategi	4	1	0	0.01055	0
okoc	okoc	4	1	0	0.01055	0
.
.
.
ISLAM	ISLAM	0	0	0	0.01055	0

LAMPIRAN 9 Output *Prediksi Naïve Bayes Classification* ketiga calon Gubernur DKI Jakarta

no	Calon	Tweet	Positif	Negatif
1	Agus	debat kali ini memgagumkan tak boleh keluar ruang orang	4.00	1.00
2	Agus	dukung amuk ht yang gw tunggu bukan menang pilkada dki tapi baper nya pepo	1.00	3.00
3	Agus	keluarga kalo kalah	1.00	2.00
.
.
.
1347	Agus	kami akan guna sistem yang akuntabel agar tidak jadi korupsi	2.00	3.00
Total			1293.00	1282.00

no	Calon	<i>Tweet</i>	Positif	Negatif
1	Ahok	pimpin jakarta itu sama dengan hubung orang tua dan anakyang ingin anak2nya sukses	3.00	1.00
2	Ahok	close statement nya pak nol	1.00	2.00
3	Ahok	closing penuh benci bagi yang tidak simpat makin tidak simpat	1.00	2.00
.	.			
.	.			
.	.			
1416	Ahok	adu ide vs soal tangan bullying di sekolah	10.00	1.00
Total			1576.00	1374.00

no	Calon	<i>Tweet</i>	Positif	Negatif
1	Anies	okoce utk cipta lap kerja di jakarta	2.00	1.00
2	Anies	jawab mas tarik tp ga nyambung re lebih paslon lain	1.00	2.00
3	Anies	ahok tanya program rumah tanpa dp	2.00	1.00
.	.			
.	.			
.	.			
1352	Anies	biasa pandai puji tapi ia susah puji paslon lain hanya bilang pandai tak ada puji	1.00	3.00
Total			1324.00	1130.00

LAMPIRAN 10 *Output Prediksi Support Vector Machine* ketiga
calon Gubernur DKI Jakarta

Kernel used:

RBF kernel: $K(x,y) = e^{-(0.01 * \langle x-y, x-y \rangle^2)}$

Classifier for classes: POSITIF, NEGATIF

BinarySMO (AHOK)

```

21.6825 * < -0.026575 -...-0.053206 -0.037596 -0.026575 -0.026575 -
0.026575 -0.06521-...-0.0460 > * X]
+   0.9515 * < -0.026575 -...-0.053206 -0.037596 -0.026575 -0.026575 -
0.026575 -0.06521-...-0.0460 > * X]
+   0.9522 * < --0.026575 -...-0.053206 -0.037596 -0.026575 -0.026575 -
0.026575 -0.06521 -...-0.0460> * X]
.
.
.
+   0.0488 * < --0.026575 -...-0.053206 -0.037596 -0.026575 -0.026575 -
0.026575 -0.06521 -...-0.0460> * X]

```

Kernel used:

RBF kernel: $K(x,y) = e^{-(0.01 * \langle x-y, x-y \rangle^2)}$

Classifier for classes: POSITIF, NEGATIF

BinarySMO (AGUS)

```

1.0236 * < - 0.047228 -0.038547 -...-0.027247 -0.054555 -0.047228 -
0.027247 -... -0.027247 > * X]
-    0.9763 * < -0.047228 -0.038547 -...-0.027247 -0.054555 -0.047228 -
0.027247 -... -0.027247 > * X]
-    0.9765 * < -0.047228 -0.038547 -...-0.027247 -0.054555 -0.047228 -
0.027247 -... -0.027247 > * X]
.
.
.
+    0.2702 * < -0.047228 -0.038547 -...-0.027247 -0.054555 -0.047228 -
0.027247 -... -0.027247 > * X]

```

Kernel used:

RBF kernel: $K(x,y) = e^{-(0.01 * \langle x-y, x-y \rangle^2)}$

Classifier for classes: POSITIF, NEGATIF

BinarySMO (ANIES)

```

1.0781 * < -0.027196 -0.038476 -...-0.027196 -0.027196 -0.027196 -
0.04714 -0.066741 -...-0.060903> * X]
+    1.0783 * < -0.027196 -0.038476 -...-0.027196 -0.027196 -0.027196 -
0.04714 -0.066741 -...-0.060903> * X]
+    0.9219 * < -0.027196 -0.038476 -...-0.027196 -0.027196 -0.027196 -
0.04714 -0.066741 -...-0.060903> * X]
.
.
.
+    1.0777 * -0.027196 -0.038476 -...-0.027196 -0.027196 -0.027196 -
0.04714 -0.066741 -...-0.060903> * X]

```

LAMPIRAN 11 Surat Pernyataan Penggunaan Data

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa
Departemen Statistika FMIPA ITS:

Nama : Eza Putra Nuansa

NRP : 1313100114

menyatakan bahwa data yang digunakan dalam Tugas Akhir/
Thesis ini merupakan data sekunder yang diambil dari ~~penelitian/~~
~~buku/~~ Tugas Akhir/ Thesis/ publikasi lainnya yaitu:

Sumber : *www.twitter.com*

Keterangan : *tweet* dari akun @AgusYudhoyono,
@basukibtp, dan @AniesBaswedan

Waktu : Masa tenang Pilgub DKI Jakarta putaran 1
(11 Februari 2017 – 14 Februari 2014)

Surat Pernyataan ini dibuat dengan sebenarnya. Apabila terdapat
pemalsuan data maka saya siap menerima sanksi sesuai aturan yang
berlaku.

Mengetahui
Pembimbing Tugas Akhir

Surabaya, Juni 2017



Dr. Kartika Hithriasari, M.Si
NIP. 196912 2 199303 2 002



Eza Putra Nuansa
NRP. 1312 100 111

*(coret yang tidak perlu)

(halaman ini sengaja dikosongkan)

BIODATA PENULIS



Eza Putra Nuansa atau yang akrab disapa Eza merupakan anak sulung dari dua bersaudara yang lahir di Jakarta, 7 Mei 1995. Putra dari pasangan Ahmad Sauji dan Nunung Budi Astuti ini berdomisili di Jakarta Selatan dan telah menempuh pendidikan dormal di SD Negeri Lenteng Agung 01 Pagi (2001-2007), SMP Negeri 98 Jakarta (2008-2010), dan SMA Negeri 109 Jakarta (2011-2013). Penulis memilih untuk melanjutkan studi guna menempuh gelar sarjana di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Pada tahun 2013, penulis dinyatakan lolos SBMPTN sebagai mahasiswa jurusan Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam. Semasa kuliah yang ditempuh dalam 4 tahun, penulis aktif di organisasi kemahasiswaan ITS tingkat jurusan yakni Himpunan Mahasiswa Statistika (HIMASTA-ITS) pada periode 2014-2015 sebagai *staff* Departemen Sosial Masyarakat dan pada periode 2015-2016 sebagai ketua departemen Sosial Masyarakat. Penulis juga aktif di forum Ikatan Himpunan Mahasiswa Statistika Indonesia sebagai Sekertaris Wilayah. Selain itu penulis turut berpartisipasi dalam kepanitian seperti ketua CERITA 2015. Di luar kegiatan kampus, penulis di semester akhir, juga sudah berkesmpatan bekerja di PT UBER Indonesia Technology, sebagai *Data Specialist*. Segala kritik dan saran serta diskusi lebih lanjut mengenai Tugas Akhir ini dapat dikirimkan melalui surat elektronik (*e-mail*) ke ezanuansa@gmail.com.